

PROPERTYX (PXON)

Deployment · Verification · Functions Report

Report Metadata

Generated at: 2026-08-07T02:17:26.050Z
Report tool: pxon-ops (Node.js, ethers v5, pdfkit)
Network: BNB Smart Chain — chainId 56
RPC endpoint: <https://bsc-dataseed1.binance.org/>

Contract Identity

Diamond address: 0x7fd59117Fe98e231a69f11C24E7FD039D360B93b
Verified contract: PropertyXDiamond (source verified on BscScan)
Compiler: Solidity 0.8.24 (vialR, optimizer runs=200, evmVersion=paris)
Creator wallet: 0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
Creation tx: 0xdc6f5eb45ac033e399da276a8cdbac3f62c28516f04d9a51f0c0b7baab781718
Token: PROPERTYX (PXON), decimals=18
totalSupply now: 5050000 PXON
Owner: 0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
Facets deployed: 65
Function selectors: 408
Resolved signatures: 397 from source + 6 from 4byte.directory
Verified on BscScan: 66 / 66 (ALL VERIFIED)

Source verification confirmed at <https://bscscan.com/address/0x7fd59117Fe98e231a69f11C24E7FD039D360B93b#code>

1. Deployment Summary

The PXON diamond was deployed by the creator wallet in a single transaction that also installed all facet cuts and executed `PXONDiamondInit.init(config)`. Below is the recovered constructor configuration decoded from on-chain calldata plus post-deployment on-chain state.

Constructor / Init parameters (from on-chain calldata)

name:	PROPERTYX
symbol:	PXON
decimals:	18
maxTotalSupply:	100,000,000 PXON (0x52b7d2dcc80cd2e400000000 wei)
usdwToken:	0xE75AD08f416D4e53e4D45dd5140A4C8b84F39Fb
treasury:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
controller:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
navOracle:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
seriesManager:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
complianceOfficer:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
porFeed:	0x00 (unset)
porMaxStaleness:	86400 s
porMinReserveRatio:	10000 bps (100%)
mintDisabledOnPoRFail:	false (warn-only)
platformFeeRate:	50 bps (0.5%)
sellSpreadBps:	50 bps
issuanceActive:	true
redeemActive:	true
complianceActive:	true
controllable:	true
gaslessEnabled:	true
maxGasPerOp:	500,000

Live on-chain snapshot

paused:	false
totalNAV:	1934262.421275698019599092
NAV / token:	0.383022261638752083
PoR reserve (E18):	1934262.4212757 (fresh=true)
Backing ratio (bps):	3830

BscScan Verification Status

Live check via Etherscan V2 unified API (chainid=56) at 2026-07-30T10:25:21.617Z.

Addresses checked: 66
Verified: 66 (ALL VERIFIED)
Unverified: 0

Every facet AND the diamond proxy has published, source-verified bytecode on BscScan. You can browse each below on BscScan by the contract name.

Per-address verification (diamond + 65 facets)

1. 0x7fd59117Fe98e231a69f11c24E7FD039D360B93b	OK	PropertyXDiamond
v0.8.24+commit.e11b9ed9 runs=200		
2. 0x08e382c26E65d29B25d3a1FeB493C080a6A9d56C	OK	DiamondCutFacet
v0.8.24+commit.e11b9ed9 runs=200		
3. 0xB0C8f800c0ECb231b5B83E9c7aa653Bd5fe10a1B	OK	DiamondLoupeFacet
v0.8.24+commit.e11b9ed9 runs=200		
4. 0x1428BC04cdd68D215cBe6FDBF178220Aafb33901	OK	OwnershipFacet
v0.8.24+commit.e11b9ed9 runs=200		
5. 0x8dEBFF25EC2d6ABb3bF73F41E5B8b5a99917DFC0	OK	ERC165Facet
v0.8.24+commit.e11b9ed9 runs=200		
6. 0x9451AB041747D86c470ad2748Ac3c285BA062E38	OK	CoreERC20FacetV4
v0.8.24+commit.e11b9ed9 runs=200		
7. 0xc8Dd6CD76dFA91b913F44807056b45fc62e52a71	OK	PermitFacet
v0.8.24+commit.e11b9ed9 runs=200		
8. 0x36b49DA6313EaCD5ef14740E2aE5782c5901576E	OK	InitERC20Facet
v0.8.24+commit.e11b9ed9 runs=200		
9. 0x54cC10A1AB6C4d1F9e925FF3967Ee90F09b95939	OK	MintBurnFacetProd
v0.8.24+commit.e11b9ed9 runs=200		
10. 0xF93A8444bBfe59eBB4fCEc0c564d045a7D50daa6	OK	MintBurnFacet
v0.8.24+commit.e11b9ed9 runs=200		
11. 0xaE55D90726860e7952f2b15151cb6e5F8f33E3A0	OK	ERC1400IssuanceFacet
v0.8.24+commit.e11b9ed9 runs=200		
12. 0x41e470e76aaBaDc2F1695CB4DCb9667EE39D5622	OK	ERC1400ComplianceFacet
v0.8.24+commit.e11b9ed9 runs=200		
13. 0xf57588C23247B57B253a93F505a9EE7D17f0259E	OK	ERC3643IdentityFacet
v0.8.24+commit.e11b9ed9 runs=200		
14. 0x0c3A9A9cAA6A7221394BE358F1003D4B101bCBd7	OK	KYCAMLFacet
v0.8.24+commit.e11b9ed9 runs=200		
15. 0x2450519E9b5c9849Db84855489DFcAf77016Cc6	OK	RolesFacet
v0.8.24+commit.e11b9ed9 runs=200		
16. 0x8e253e21DdeE13aA365a11C72816eC0Bc75DB3FD	OK	RoleAdminActionsFacet
v0.8.24+commit.e11b9ed9 runs=200		
17. 0x3065738c032c0605C363830423d2756fB192c45F	OK	ERC1400PartitionFacet
v0.8.24+commit.e11b9ed9 runs=200		
18. 0x77cE2eB410c74bB81D36A9553434Caa985c6527a	OK	RedemptionFacet
v0.8.24+commit.e11b9ed9 runs=200		
19. 0x4d422BA36DF9229A3eEC62785444D36561EE160d	OK	RedemptionVaultFacet
v0.8.24+commit.e11b9ed9 runs=200		
20. 0x129bD97b4C8B95E318e551bEfe714c54D9581618	OK	PorFacet
v0.8.24+commit.e11b9ed9 runs=200		
21. 0x2719eb6334070DFf2a8B30554e68bDBa37992DF8	OK	PorSnapshotFacet
v0.8.24+commit.e11b9ed9 runs=200		
22. 0x920e38fda9665Eb7E6849441a056822EB755944f	OK	PorPolicyFacet
v0.8.24+commit.e11b9ed9 runs=200		
23. 0x0CF0c53fa3273cdF7D4B932CD1eb5ED8dD38CDe6	OK	PorMultiChainFacet
v0.8.24+commit.e11b9ed9 runs=200		
24. 0xd3d6aAec0F3685cf0ccEE189aE6F3dfc48447635	OK	SeriesEngineFacet
v0.8.24+commit.e11b9ed9 runs=200		
25. 0xe80901Fa529870F2797257C57Ef015608FB1dCcB	OK	ERC1400DocumentFacet
v0.8.24+commit.e11b9ed9 runs=200		
26. 0xe6F1373Dc5504208388cc35924E4009a56af4099	OK	ERC1400ControllerFacet
v0.8.24+commit.e11b9ed9 runs=200		
27. 0xB51C068E1913193467F8D7cA930D34CEF38Be9c1	OK	PriceFeedFacetBSC
v0.8.24+commit.e11b9ed9 runs=200		
28. 0x8aA6c2716FC28E09bE631A344952E8dB380A6bb9	OK	CanonicalOracleFacet
v0.8.24+commit.e11b9ed9 runs=200		
29. 0x83197BD5f95a18e2cF83ab26fD417e62581a6cc4	OK	OracleViewFacet
v0.8.24+commit.e11b9ed9 runs=200		
30. 0xE861cE800fD38fd466b1dF1137cCbdb82DB2d177e	OK	PriceLogicFacet
v0.8.24+commit.e11b9ed9 runs=200		
31. 0x8a4e8E5926c049528A3a3c43053084c952A1a848	OK	EmergencyFacet
v0.8.24+commit.e11b9ed9 runs=200		
32. 0x06785aB20E3D8c911B27827035325064Cfb4109f	OK	ComplianceFacet
v0.8.24+commit.e11b9ed9 runs=200		
33. 0x199c5F82C01F3ff796BFb4107eF0dEda8a271231	OK	PegPolicyFacet
v0.8.24+commit.e11b9ed9 runs=200		
34. 0xE48A27d48c1555b494fEEFA13d62cfc8494EACf1	OK	UniversalPegGuard
v0.8.24+commit.e11b9ed9 runs=200		
35. 0x560c6383ab87d0ffCF155A79d1b99dc28E7484cD	OK	ISOControlFacet
v0.8.24+commit.e11b9ed9 runs=200		
36. 0x7D2Cd5889640934C3f84265f9043160e7798878a	OK	StakingPluginFacet
v0.8.24+commit.e11b9ed9 runs=200		

37. 0xB20ecf82AaC86D8E542cbb7b6c2c2d668caB69CD OK
v0.8.24+commit.e11b9ed9 runs=200
38. 0x7317603FfB38F6625dB191F50Ce3411056EE6b0A OK
v0.8.24+commit.e11b9ed9 runs=200
39. 0xF24958ADc961c9Dd47372DE6424cF2d1637b52C1 OK
v0.8.24+commit.e11b9ed9 runs=200
40. 0xba8dB0ad3db771224D25D3AFab22E04b95649667 OK
v0.8.24+commit.e11b9ed9 runs=200
41. 0xbF71ce399A8731DB4A1A417D66791AA5f7Ef76B1 OK
v0.8.24+commit.e11b9ed9 runs=200
42. 0x1c8bF6224D4d70873D7401a3EbB2F6dA01762246 OK
v0.8.24+commit.e11b9ed9 runs=200
43. 0x8559c79b0D02f123c8945975beA8322cC9C5C27A OK
v0.8.24+commit.e11b9ed9 runs=200
44. 0xE341C12F82e3784fC4A30f01DC8c41E00bE966ab OK
v0.8.24+commit.e11b9ed9 runs=200
45. 0x69110e5f087f2910257A87dAFBb7a3f59b013b2A OK
v0.8.24+commit.e11b9ed9 runs=200
46. 0x08F1F90AE66C50cF32672d505708cB108981dA18 OK
v0.8.24+commit.e11b9ed9 runs=200
47. 0x7E4283246aa10E97b768fcF8Ae1120299A990bD4 OK
v0.8.24+commit.e11b9ed9 runs=200
48. 0xd09fD383f64A71c6f79409c2b3f85F4C23212961 OK
v0.8.24+commit.e11b9ed9 runs=200
49. 0x7050dC6E4e5a1B84EbDff683E3B3236679E56351 OK
v0.8.24+commit.e11b9ed9 runs=200
50. 0x66017098AE601bb49e47581b81CA000310968aac OK
v0.8.24+commit.e11b9ed9 runs=200
51. 0x9917D42224bbD227F0E48ddC020590248322287a OK
v0.8.24+commit.e11b9ed9 runs=200
52. 0x5e725c33A2319339Bf9d290144f944F3Fd4CD4aD OK
v0.8.24+commit.e11b9ed9 runs=200
53. 0xFD828c1cc4465B3aB9792C44ac825Ad491cFD13D OK
v0.8.24+commit.e11b9ed9 runs=200
54. 0xF50d3d52E64ca53D8c70c1A557D28461546A59F5 OK
v0.8.24+commit.e11b9ed9 runs=200
55. 0x81a0B2592bcfB03AcB6C9dB9a082C11b81780700 OK
v0.8.24+commit.e11b9ed9 runs=200
56. 0xeAa3602B855e8d06452f42F77a9D1F3d5486CEeb OK
v0.8.24+commit.e11b9ed9 runs=200
57. 0x203f453D7343DED6Ef035b6fCe4ca7943C0ee6BC OK
v0.8.24+commit.e11b9ed9 runs=200
58. 0xf5Cb0ec8392769F52F051a27641D91541AAF8206 OK
v0.8.24+commit.e11b9ed9 runs=200
59. 0x7B40D027227b245ED9cF84B64E952b8d95797898 OK
v0.8.24+commit.e11b9ed9 runs=200
60. 0xea388a994caB9a1674B84e009bfB2d5a78B9BDD1 OK
v0.8.24+commit.e11b9ed9 runs=200
61. 0xc57A4415C7e365712E546987def28B59082C127B OK
v0.8.24+commit.e11b9ed9 runs=200
62. 0xc4b7FbfF19F18237bD4E284ba2C921182e7b7f62 OK
v0.8.24+commit.e11b9ed9 runs=200
63. 0xF8075E26166a14E3412FBe24643732f7678dbc50 OK
v0.8.24+commit.e11b9ed9 runs=200
64. 0xf53040B37f011Af12831fd74AD6Ee57a70343b31 OK
v0.8.24+commit.e11b9ed9 runs=200
65. 0x553DaCe32909730e1851Fe55331D178Ee79f655A OK
v0.8.24+commit.e11b9ed9 runs=200
66. 0xb68aC8F3AF87Ad180db45ACeA9765c24493aFf73 OK
v0.8.24+commit.e11b9ed9 runs=200

LockingFacet
VestingPluginFacet
GovernanceFacet
FeatureFlagsFacet
ConfigFacet
AuditBoardFacet
AdminUtilsFacet
FXRatesFacet
MultiCurrencyReserveFacet
GaslessFacet
TokenIdentityFacet
CrossChainRegistryFacet
MarketDataFacet
UpgradeTimelockFacet
RebasePluginFacet
AuthenticityGuardFacet
CatalogComplianceFacet
GovernorAdapterFacet
BridgePluginFacet
CrossChainAdapterFacet
BridgeCompatFacet
AdapterRegistryFacet
GatewayRegistryFacet
VoucherFacet
UpgradeabilityCompatFacet
PegReferenceFacet
AIOracleFacet
PriceSourcesFacet
RebaseElasticFacet
RealDataFacet

The following mint was executed against the live diamond.

tx hash:	0xb2a432dda0ff362e09327fdf376a4c10e845110eecb98a4c382d5d20bb72e2c8
block:	112993901
signer:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
recipient:	0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6
amount:	100000000 PXON (1000000000000000000000000 wei)
gas used:	81860
effective gasPrice:	50000000
status:	success (1)
post totalSupply:	100000000 PXON
post recipient bal:	100000000 PXON

4. Plot Registry (Land-Lot Series)

Each land lot is registered on-chain via `SeriesEngineFacet.createSeries(...)`. The `plotNumber` is stored in the `SeriesMeta.name` field. Cadastral doc hashes go in `assetHash/docRoot`.

Template lots: 98
PXON represented: 100000000 PXON
Source file: pxon-ops\plots.json

Sample entries (first 20)

LIFE-SKL-A16-105825 issuance=0.019264983621139667	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=36854.416139856570225675
LIFE-SKL-A17-105826 issuance=0.019273381608770394	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=35665.564006312808487564
LIFE-SKL-A18-105827 issuance=0.019309372984324179	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=34476.711872769046749454
LIFE-SKL-A19-105828 issuance=0.019348084638262847	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=33287.859739225285011344
LIFE-SKL-A20-105829 issuance=0.019389837086770716	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=32099.007605681526911212
LIFE-SKL-A21-105830 issuance=0.01929645797563919	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=33585.072772611223626882
LIFE-SKL-A22-105831 issuance=0.019383027901106468	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=38340.481306786270579323
LIFE-SKL-A23-105832 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A24-105833 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A25-105834 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A26-105835 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A27-105836 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A28-105837 issuance=0.019375957029091201	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=28829.664238436187588377
LIFE-SKL-A29-105838 issuance=0.019304530508336838	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=39826.546473715970932972
LIFE-SKL-B30-105839 issuance=0.019226142928362649	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=19616.060203472043212969
LIFE-SKL-B31-105840 issuance=0.019355981815659149	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=16941.142902998584759189
LIFE-SKL-B32-105841 issuance=0.019355981815659149	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=16941.142902998584759189
LIFE-SKL-B33-105842 issuance=0.019355981815659149	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=16941.142902998584759189
LIFE-SKL-B34-105843 issuance=0.019355981815659149	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=16941.142902998584759189
LIFE-SKL-B35-105844 issuance=0.019355981815659149	spv=0xc557ee55f5DbDAc7128fc4D378111a7D68dEb3d6	navValue=16941.142902998584759189

... 78 more not shown.

On-chain registration status

Registered lots: 1
Successful: 1
Failed: 0

Per-lot receipts (first 30)

LIFE-SKL-B79-105888 id=98 tx=0xb74e69a1e907d2ca979b6097222524b785fa23d94e70ec79611804f3ad43320a

5. Proof-of-Reserve & NAV Snapshot

Values below are read live from the diamond via PorFacet at the time this report was generated.

totalNAV (USDW-18):	1934262.421275698019599092
NAV / PXON token:	0.383022261638752083
Reserve (E18):	1934262.4212757
Updated at (unix):	1785476192
isFresh:	true
Backing ratio (bps):	3830

6. Full Function / Selector Inventory

Every selector below is currently routed by the diamond (`LoupeFacet.facets()`). Signatures resolved from local Solidity sources are marked `[src]`, those recovered from `4byte.directory` are `[4b]`, and unresolved selectors are shown as `bytes4` only.

Facet 0x9451AB041747D86c470ad2748Ac3c285BA062E38 (22 selectors)

0xdd62ed3e	[src]	allowance(address,address)
0x095ea7b3	[src]	approve(address,uint256)
0x70a08231	[src]	balanceOf(address)
0x313ce567	[src]	decimals()
0xa457c2d7	[src]	decreaseAllowance(address,uint256)
0x39509351	[src]	increaseAllowance(address,uint256)
0xd4b53324	[src]	killerKillSwitch()
0xd5abeb01	[src]	maxSupply()
0x06fdde03	[src]	name()
0x6af0313d	[src]	pegEnforcer()
0x7d6ba9d8	[src]	pegKillSwitch()
0x4cfd017e	[src]	setKillerKillSwitch(bool)
0x68a394cf	[src]	setPegEnforcer(address)
0xfa9866a4	[src]	setPegKillSwitch(bool)
0x1b8ae643	[src]	setShieldKillSwitch(bool)
0xd4bd8ba2	[src]	shieldKillSwitch()
0x95d89b41	[src]	symbol()
0x18160ddd	[src]	totalSupply()
0xa9059cbb	[src]	transfer(address,uint256)
0x23b872dd	[src]	transferFrom(address,address,uint256)
0xee532f31	[src]	transferFromWithData(address,address,uint256,bytes)
0x2535f762	[src]	transferWithData(address,uint256,bytes)

Facet 0x41e470e76aaBaDc2F1695CB4DCb9667EE39D5622 (19 selectors)

0x8eda1889	[src]	activateCompliance(bool,bool)
0x3643adb5	[src]	batchSetKYC(address[],bool[])
0x1badb25c	[src]	canTransfer(address,uint256,bytes)
0x9945e70e	[src]	canTransferByPartition(address,address,bytes32,uint256,bytes)
0x122eb575	[src]	canTransferFrom(address,address,uint256,bytes)
0x5b090ed6	[src]	complianceStatus()
0x1615f2d4	[src]	getRiskLevel(address)
0x5edf9e61	[src]	getTransferLimit(address)
0xfe575a87	[src]	isBlacklisted(address)
0xe77db232	[src]	isKYCApproved(address)
0xc14d1e4c	[src]	isRedFlagged(address)
0x3af32abf	[src]	isWhitelisted(address)
0xd01dd6d2	[src]	setBlacklisted(address,bool)
0xe7eeff76	[src]	setGlobalLimits(uint256,uint256)
0x6e72a6bc	[src]	setKYCApproved(address,bool)
0xe8e5a085	[src]	setRedFlagged(address,bool)
0x67315b3e	[src]	setRiskLevel(address,uint8)
0x953715ae	[src]	setTransferLimit(address,uint256,uint256)
0x9281aa0b	[src]	setWhitelisted(address,bool)

Facet 0xf57588C23247B57B253a93F505a9EE7D17f0259E (18 selectors)

0x9c5db6d9	[src]	batchRegisterIdentities(address[],address[],uint16[])
0xa8d29d1d	[src]	deleteIdentity(address)
0xc79339e2	[src]	getComplianceModule()
0x5ad0c0bf	[src]	getIdentityRecord(address)
0x3e54e07e	[src]	getSequencerFeed()
0xf0eb5e54	[src]	identity(address)
0xd7e64c00	[src]	investorCount()
0x7e42683b	[src]	investorCountry(address)
0x05abe2ec	[src]	isCountryBlocked(uint16)
0xc3c5a547	[src]	isRegistered(address)
0xb9209e33	[src]	isVerified(address)
0x454a03e0	[src]	registerIdentity(address,address,uint16)
0x423db9c7	[src]	setComplianceModule(address)
0x8db7b007	[src]	setCountryBlocked(uint16,bool)
0x35e896c1	[src]	setMaxInvestors(uint256)
0x1dabe830	[src]	setSequencerFeed(address,uint256)
0x3b239a7f	[src]	updateCountry(address,uint16)
0x8e098ca1	[src]	updateIdentity(address,address)

Facet 0x8aA6c2716FC28E09bE631A344952E8dB380A6bb9 (18 selectors)

0x7542f23a	[src]	pegPrice()
0x18ff3c8b	[src]	pokePeg()
0xb95ed06f	[src]	priceOf(address)
0xdd47efb7	[src]	REQUIRED_DECIMALS()
0xacbbca7b	[src]	REQUIRED_DESCRIPTION()
0xe60211dc	[src]	setQuorum(uint16)
0x63528565	[src]	setSource(uint8,address,string)
0xb893fe85	[src]	sourceLabel(uint8)
0x7df153d1	[src]	source(uint8)
0xf914f712	[src]	usdwLatestAnswer()

0x30dd6a6d	[src]	usdwLatestRound()
0xf3e316d4	[src]	usdwLatestRoundData()
0xaaee87a07	[src]	usdwLatestTimestamp()
0xc1bc8ad9	[src]	usdwOracleDecimals()
0xe5c50828	[src]	usdwOracleDescription()
0x0ffc1a32	[src]	usdwOracleVersion()
0xa9ed7c36	[?]	(unresolved)
0xba7b06ff	[?]	(unresolved)

Facet 0x560c6383ab87d0ffCF155A79d1b99dc28E7484cD (17 selectors)

0x6147c173	[src]	anchorEvidence(bytes32,bytes32)
0x71020dff	[src]	checkRedFlagGuard(bytes4,address,address)
0x99737793	[src]	continuityState()
0x2eb99313	[src]	emitWebhook(bytes32,bytes)
0x4a7221a0	[src]	getEvidence(bytes32)
0x7f9e0fd2	[src]	getObligation(bytes32)
0x78e785ad	[src]	isRedFlag(address)
0x4b9f9fbf	[src]	providerApproved(bytes32)
0xf50bf174	[src]	registerObligation(bytes32,bytes32)
0x5d6dfa9e	[src]	setContinuityMode(uint8,bytes32)
0x521fcd32	[src]	setFunctionPaused(bytes4,bool,bytes32)
0xf7fb3e42	[src]	setGlobalHalt(bool,bytes32)
0x87325830	[src]	setProviderApproved(bytes32,bool)
0x7814103d	[src]	setRedFlag(address,bool,bytes32)
0x67febe72	[src]	setSignerApproved(address,bool)
0xcb4a4def	[src]	signerApproved(address)
0x6d9b99eb	[src]	webhookNonce(bytes32)

Facet 0xd3d6aAec0F3685cf0ccEE189aE6F3dfC48447635 (14 selectors)

0x9bccdf41	[src]	buySeriesTokens(uint256,uint256)
0x9219e510	[src]	cancelRedemption(uint256)
0x948b5165	[src]	createSeries(string,address,uint256,uint256,uint256,uint256,uint256,bytes32,bytes32)
0xeac1ee47	[src]	exitSeries(uint256)
0x75deb2a7	[src]	fulfillRedemption(uint256)
0xf7d57f1c	[src]	getInvestorRedemptions(address)
0xf5ceb211	[src]	getSeriesBalance(address,uint256)
0x4510825f	[src]	getSeriesInvestors(uint256)
0x1160da15	[src]	getSeries(uint256)
0x14b9abc9	[src]	sellSeriesTokens(uint256,uint256)
0xd7f2c0ef	[src]	seriesCount()
0xb0a9b406	[src]	setSeriesFrozen(uint256,bool)
0x7f407edf	[src]	setSpread(uint256,uint256)
0xb50614a8	[src]	updateSeriesDocRoot(uint256,bytes32,bytes32)

Facet 0xB51C068E1913193467F8D7cA930D34CEF38Be9c1 (14 selectors)

0x9358928b	[src]	circulatingSupply()
0x7284e416	[src]	description()
0xb86e7822	[src]	feedSummary()
0xd7355955	[src]	isPegged()
0x50d25bcd	[src]	latestAnswer()
0xfeaf968c	[src]	latestRoundData()
0xb8f01c3b	[src]	liquidityUSD()
0x52d3f41a	[src]	marketCapUSD()
0x9a105d2b	[src]	pegTargetE18()
0x6301a747	[src]	priceFeedVersion()
0x93a408d7	[src]	priceUSD()
0xfd915e6d	[src]	reserveBacking()
0x5853b613	[src]	touchFeed()
0x54fd4d50	[src]	version()

Facet 0xb68aC8F3AF87Ad180db45ACeA9765c24493aFf73 (14 selectors)

0x80e0704b	[src]	cmcSlug()
0xc2fe0fd8	[src]	coingeckoId()
0x4ad17cdc	[src]	getPriceHistory(uint16)
0xb745ae93	[src]	getRealMetrics()
0x7406f76b	[src]	initRealData(address,address,string,string)
0x12d0b57c	[src]	recordSwap(address,address,uint256,bool)
0xf968dfdd	[src]	setCMCSlug(string)
0x4a2f7d73	[src]	setCoingeckoId(string)
0xc30ff6df	[src]	setLogoURI(string)
0xa126f13a	[src]	snapshotHourly()
0x8820a08e	[src]	tokenLogoURI()
0x6c02a931	[src]	tokenName()
0x7b61c320	[src]	tokenSymbol()
0x3defeb04	[src]	tokenWebsite()

Facet 0x3065738c032c0605C363830423d2756fB192c45F (13 selectors)

0x103ef9e1	[src]	authorizeOperatorByPartition(bytes32,address)
0x30e82803	[src]	balanceOfByPartition(bytes32,address)
0x6d77cad6	[src]	isOperatorForPartition(bytes32,address,address)
0xa1422f69	[4b]	LOCKED()
0x8c0dee9c	[src]	operatorTransferByPartition(bytes32,address,address,uint256,bytes,bytes)
0x740ab8f4	[src]	partitionsOf(address)
0x9d2cc436	[4b]	RESERVE()
0x168ecec5	[src]	revokeOperatorByPartition(bytes32,address)
0x069c6631	[src]	seedPartitionBalance(address,bytes32,uint256)
0x69598efe	[src]	totalPartitions()

0xf3d490db	[src]	transferByPartition(bytes32,address,uint256,bytes)
0x6c2f0d53	[4b]	UNLOCKED()
0x40f9f60e	[?]	(unresolved)

Facet 0x08F1F90AE66C50cF32672d505708cB108981dA18 (12 selectors)

0xc0890814	[src]	batchApproveTargets(address[],bool[])
0xbb44d15c	[src]	executeGasless(address,address,bytes,uint256,uint256,uint256,uint8,bytes32,bytes32)
0xa530d833	[src]	gaslessNonce(address)
0xae52c569	[src]	getApprovedTargets()
0xf47c5cdb	[src]	isApprovedTarget(address)
0x963a959e	[src]	isGaslessEnabled()
0x52636ebe	[src]	isOpExecuted(bytes32)
0x1b2d6577	[src]	maxGasPerOp()
0xb7f9a748	[src]	setApprovedTarget(address,bool)
0x395c3226	[src]	setGasless(bool)
0x3a140e38	[src]	setMaxGasPerOp(uint256)
0x2c985cff	[?]	(unresolved)

Facet 0x129bD97b4C8B95E318e551bEfe714c54D9581618 (11 selectors)

0xf335d56a	[src]	batchUpdateNAV(uint256[],uint256[])
0x18b9be2e	[src]	checkMintPolicy(uint256)
0x90f48062	[src]	getBackingRatioBps()
0xa291a842	[src]	getNAVPerToken()
0x54ed0253	[src]	getPoRReserve()
0xe92fcab2	[src]	getSeriesNAVPerToken(uint256)
0x1a9ade51	[src]	porFeedAddress()
0x9b8417b8	[src]	porMaxStaleness()
0xedd129a9	[src]	setPorFeed(address,uint256)
0x30421927	[src]	totalNAV()
0x06c83f1c	[src]	updateSeriesNAV(uint256,uint256)

Facet 0x2719eb6334070DFf2a8B30554e68bDBa37992DF8 (10 selectors)

0x0e0b0db6	[src]	getCertFeed()
0xde8c641	[src]	getSnapshotTimestamp()
0x76f10ad0	[src]	getSnapshot(uint256)
0x4aafbddb	[src]	isSnapshotFresh(uint256)
0xcb22af4c	[src]	latestSnapshot()
0xa1dae74b	[src]	porSnapshotHash()
0x795d5bb4	[src]	setCertFeed(address)
0xc137ec56	[src]	setCertFeedMaxStaleness(uint256)
0x534fbe98	[src]	setPorSnapshotHash(bytes32)
0x098ab6a1	[src]	snapshotCount()

Facet 0x83197BD5f95a18e2cF83ab26fD417e62581a6cc4 (10 selectors)

0x63929d58	[src]	CHAINLINK_PEG_ANSWER()
0x3f903940	[src]	feedMeta()
0x16da057b	[src]	getLastVerified()
0xaecde588	[src]	getMaxStaleness()
0x833b1fce	[src]	getOracle()
0x98d5fdca	[src]	getPrice()
0x8ba83c0b	[src]	PEG_USD()
0x335e077a	[src]	refreshAndVerify()
0x100bc9e1	[src]	setMaxStaleness(uint256)
0x7adbf973	[src]	setOracle(address)

Facet 0x7E4283246aa10E97b768fcF8Ae1120299A990bD4 (10 selectors)

0xeb6db8f5	[src]	addChainDeployment(uint256,string,address,uint32)
0xe8a3d485	[src]	contractURI()
0xd25d23bd	[src]	getChainCount()
0xd7f24abd	[src]	getChainDeployment(uint256)
0x0a8f41b4	[src]	getSocialLinks()
0x995d9ab7	[src]	getTags()
0x7b96b8e0	[src]	getTokenIdentity()
0xae5a1ab	[src]	initTokenIdentity(string,address)
0x0570df66	[src]	setTokenLogo(string)
0x291526ac	[src]	tokenLogo()

Facet 0x0CF0c53fa3273cdF7D4B932CD1eb5ED8dD38CDe6 (9 selectors)

0x1483a234	[src]	chainReserveUSDE18(uint256)
0x1d29777b	[src]	getMultiChainConfig()
0xf6be0afe	[src]	getReserveSnapSafe(uint256,address)
0x78c8f674	[src]	getReserveSnap(uint256,address)
0xec7c9607	[src]	setChainReserveUSDE18(uint256,uint256,bytes32,uint64)
0xdc6e1022	[src]	setMultiChainConfig(uint256,uint256)
0x96bb9da5	[src]	setReserveSnap(uint256,address,uint256,bytes32,uint64)
0x4c4bf9ea	[src]	totalReserveTokenPairsUSDE18(uint256[],address[])
0x688595a0	[src]	totalReserveUSDE18(uint256[])

Facet 0xe80901Fa529870F2797257C57Ef015608FB1dCcB (9 selectors)

```
0xba69cb9b [src] batchSetDocuments(bytes32[],string[],bytes32[])
0x15a4923c [src]
bootstrapDocuments(string,bytes32,string,bytes32,string,bytes32,string,bytes32,string,bytes32)
0xcbfe6d41 [src] documentExists(bytes32)
0x9fa5f50b [src] getAllDocuments()
0xb10d6b41 [src] getDocument(bytes32)
0xf4e0c809 [src] getSeriesDocument(uint256,bytes32)
0xc3501848 [src] removeDocument(bytes32)
0x010648ca [src] setDocument(bytes32,string,bytes32)
0xd7b84c9b [src] setSeriesDocument(uint256,bytes32,string,bytes32)
```

Facet 0x4d422BA36Df9229A3eEC62785444D36561EE160d (8 selectors)

```
0xf0920278 [src] depositUSDW(uint256)
0xa30c7bb0 [src] getRedeemRecord(bytes32)
0xf1fb0036 [src] initRedeemVault(uint256,uint256,uint256)
0xec143adc [src] setMaxRedeemPerDay(uint256)
0x1253b004 [src] setMaxRedeemPerTx(uint256)
0xc2f07cc5 [src] setMinCollateralBps(uint256)
0x6ff4ab8a [src] vaultStats()
0x288e5c58 [src] withdrawUSDW(address,uint256)
```

Facet 0x8a4e8E5926c049528A3a3c43053084c952A1a848 (8 selectors)

```
0xb1e3268b [src] emergencyStop(string)
0xb187bd26 [src] isPaused()
0x7faf6a79 [src] lastPause()
0x39c09ef5 [src] lastUnpause()
0xcd11d1d2 [src] pauseEx(uint8,string)
0x60732533 [src] pause(uint8,string)
0x8893172e [src] unpauseEx(uint8,string)
0xb6d9fa09 [src] unpause(uint8,string)
```

Facet 0x7D2Cd5889640934C3f84265f9043160e7798878a (8 selectors)

```
0x372500ab [src] claimRewards()
0xe29a9370 [src] isStakingEnabled()
0x31d7a262 [src] pendingRewards(address)
0x9e447fc6 [src] setRewardRate(uint256)
0xbce8567e [src] setStakingEnabled(bool)
0x16765391 [src] stakedBalanceOf(address)
0xa694fc3a [src] stake(uint256)
0x2e17de78 [src] unstake(uint256)
```

Facet 0xea388a994caB9a1674B84e009bfB2d5a78B9BDD1 (8 selectors)

```
0x827ba03a [src] claimVoucher(bytes32)
0x9d11c3ac [src] createVoucher(uint256,uint256,bytes32)
0xf24f50bc [src] getDefaultVoucherExpiry()
0x1a96e8bc [src] getVouchersByCreator(address)
0xf175075a [src] isVoucherActive(bytes32)
0x6f486781 [src] sweepExpiredVoucher(bytes32)
0x439f30a4 [src] totalVouchersCreated()
0x29ad0d05 [src] totalVoucherValue()
```

Facet 0x920e38fda9665Eb7E6849441a056822EB755944f (7 selectors)

```
0x61458b4e [src] checkMintPolicy(uint256,uint256)
0x7ca7a971 [src] enforceMintPolicy(uint256,uint256)
0xc6dcc97c [src] getLastPolicyCheck()
0xdb249f10 [src] getMinReserveRatio()
0x58a41a9c [src] isMintingDisabledOnReserveFailure()
0x83685135 [src] setMinReserveRatio(uint256)
0xbd8f3f02 [src] setMintingDisabledOnReserveFailure(bool)
```

Facet 0xe6F1373Dc5504208388cc35924E4009a56af4099 (7 selectors)

```
0xf77c4791 [src] controller()
0xe02dbcb5 [src] controllerActionCount()
0x2bc6acc3 [src] controllerRedeem(address,uint256,bytes,bytes)
0xf282527a [src] controllerTransfer(address,address,uint256,bytes,bytes)
0x4f4bad4e [src] getControllerAction(uint256)
0x4c783bf5 [src] isControllable()
0x92eefe9b [src] setController(address)
```

Facet 0xE48A27d48c1555b494fEEFA13d62cfC8494EACf1 (7 selectors)

```
0x8e40b35e [src] enforceInvariant(address,address,uint256)
0x2cedb376 [src] getGuardConfig()
0x77f62e04 [src] initUniversalGuard(address,address,uint256,uint256)
0x48334596 [src] isAMMAAddress(address)
0x3ea6e071 [src] setExemption(address,bool)
0x2e3cfe49 [src] setGuardPaused(bool)
0x703b61cb [src] setMaxBandBps(uint256)
```

Facet 0xF50d3d52E64ca53D8c70c1A557D28461546A59F5 (7 selectors)

0x40e58ee5	[src]	cancel(uint256)
0xfe0d94c1	[src]	execute(uint256)
0x0c340a24	[src]	governor()
0x37558af5	[src]	propose(bytes)
0xddf0b009	[src]	queue(uint256)
0xc42cf535	[src]	setGovernor(address)
0x943e8216	[src]	vote(uint256,uint8)

Facet 0xaE55D90726860e7952f2b15151cb6e5F8f33E3A0 (6 selectors)

0x2f1cae85	[src]	isIssuable()
0x36d948d1	[src]	issuanceNonce(address)
0xbb3acde9	[src]	issue(address,uint256,bytes)
0x9675193c	[src]	redeemFrom(address,uint256,bytes)
0xe77c646d	[src]	redeem(uint256,bytes)
0x9ab309ca	[src]	setIssuable(bool)

Facet 0xE861cE800fD38fd466b1dF1137cCbd82DB2d177e (6 selectors)

0xc97ab421	[src]	addPriceSource(address)
0xb7f88320	[src]	getLastUpdateTime()
0xbb2ec9af	[src]	getPriceSources()
0x78bc2ff9	[src]	isPriceSource(address)
0x623b6549	[src]	removePriceSource(address)
0x8d6cc56d	[src]	updatePrice(uint256)

Facet 0x36b49DA6313EaCD5ef14740E2aE5782c5901576E (5 selectors)

0xc380a285	[src]	initERC20(string,string,uint8)
0x7438b937	[src]	setDomainSeparator(bytes32)
0xf2cf47be	[src]	setTokenDecimals(uint8)
0xa4f29aad	[src]	setTokenName(string)
0xba51b1b4	[src]	setTokenSymbol(string)

Facet 0x199c5F82C01F3ff796BFb4107eF0dEDa8a271231 (5 selectors)

0x62e0bdbe	[src]	assertPegE18(uint256)
0x3e69fa76	[src]	initPegPolicyDefaults()
0x0f64d73e	[src]	isPegHealthyE18(uint256)
0x7cf981f2	[src]	pegMinE18()
0x4477324b	[src]	setPegPolicyE18(uint256,uint256)

Facet 0xFD828c1cc4465B3aB9792C44ac825Ad491cFD13D (5 selectors)

0xe0cdb2d9	[src]	categoryEnabled(uint8)
0x02c5d006	[src]	categoryMask()
0xf961d8b7	[src]	functionAllowed(bytes4)
0x4e29fe09	[src]	setCategoryEnabled(uint8,bool)
0xa96150df	[src]	setSelectorPolicy(bytes4,bool)

Facet 0xB0C8f800c0ECb231b5B83E9c7aa653Bd5fe10a1B (4 selectors)

0xcdffacc6	[src]	facetAddress(bytes4)
0x52ef6b2c	[src]	facetAddresses()
0xadfca15e	[src]	facetFunctionSelectors(address)
0x7a0ed627	[src]	facets()

Facet 0xc8Dd6CD76dFA91b913F44807056b45fc62e52a71 (4 selectors)

0x3644e515	[src]	DOMAIN_SEPARATOR()
0x7ecele00	[src]	nonces(address)
0xd505accf	[src]	permit(address,address,uint256,uint256,uint8,bytes32,bytes32)
0x30adf81f	[4b]	PERMIT_TYPEHASH()

Facet 0x2450519E9b5c9849DbA84855489DFcAf77016Cc6 (4 selectors)

0x248a9ca3	[src]	getRoleAdmin(bytes32)
0x2f2ff15d	[src]	grantRole(bytes32,address)
0x91d14854	[src]	hasRole(bytes32,address)
0xd547741f	[src]	revokeRole(bytes32,address)

Facet 0x06785aB20E3D8c911B27827035325064Cfb4109f (4 selectors)

0x8456cb59	[src]	pause()
0x5c975abb	[src]	paused()
0xc4590d3f	[src]	setLimits(uint256,uint256)
0x3f4ba83a	[src]	unpause()

Facet 0xB20ecf82AaC86D8E542cbb7b6c2c2d668caB69CD (4 selectors)

0x929ec537	[src]	getLockedAmount(address)
0xdbed1e88	[src]	getUnlockTime(address)
0xe2ab691d	[src]	lock(address,uint256,uint256)
0x2f6c493c	[src]	unlock(address)

Facet 0x7317603Fb38F6625dB191F50Ce3411056EE6b0A (4 selectors)

```
0x5e60bf7b [src] createVesting(address, uint256, uint64, uint64, uint64)
0xe4bf01a8 [src] releasable(uint256)
0x37bdc99b [src] release(uint256)
0x2f5c4fe1 [src] setVestingEnabled(bool)
```

Facet 0xF24958ADc961c9Dd47372DE6424cF2d1637b52C1 (4 selectors)

```
0x5c19a95c [src] delegate(address)
0x8d22ea2a [src] delegateOf(address)
0x587cde1e [src] delegates(address)
0x9ab24eb0 [src] getVotes(address)
```

Facet 0xbF71ce399A8731DB4A1A417D66791AA5f7Ef76B1 (4 selectors)

```
0x21f8a721 [src] getAddress(bytes32)
0x6dd5b69d [src] getConfig(bytes32)
0xca446dd9 [src] setAddress(bytes32, address)
0x15fe96dc [src] setConfig(bytes32, uint256)
```

Facet 0xE341C12F82e3784fC4A30f01DC8c41E00bE966ab (4 selectors)

```
0x630ed513 [src] getFXRate(bytes32)
0x9871aa6e [src] quoteCurrencyToUSDWE18(uint256, bytes32)
0x0a503bab [src] quoteUSDWToCurrencyE18(uint256, bytes32)
0x53a24645 [src] setFXRate(bytes32, uint256, uint64)
```

Facet 0x7050dC6E4e5a1B84EbDff683E3B3236679E56351 (4 selectors)

```
0x4d96264c [src] getMarketData()
0xcd238f76 [src] getMarketDataFull()
0x2f4a83e7 [src] marketDataVersion()
0x507befa8 [src] priceUsdFloat()
```

Facet 0x9917D42224bbD227F0E48ddC020590248322287a (4 selectors)

```
0x4162640b [src] indexE18()
0x7c2d5327 [src] isRebaseEnabled()
0xca145b53 [src] setIndexE18(uint256)
0x4b56de38 [src] setRebaseEnabled(bool)
```

Facet 0x81a0B2592bcfB03AcB6C9dB9a082C11b81780700 (4 selectors)

```
0x69c366fe [src] burnForBridge(uint256, uint256, bytes32)
0x0c700d71 [src] setBridgeEnabled(bool)
0x20295abe [src] setBridgeValidator(address, bool)
0x0d35bd66 [?] (unresolved)
```

Facet 0xf5Cb0ec8392769F52F051a27641D91541AAF8206 (4 selectors)

```
0x9d2a954d [src] getAdapter(bytes32, bytes32)
0xc5d16c6c [src] getHook(bytes32, bytes32)
0x7b34de77 [src] setAdapter(bytes32, bytes32, address)
0x0face053 [src] setHook(bytes32, bytes32, bytes32)
```

Facet 0x1428BC04cdd68D215cBe6FDBF178220Aafb33901 (3 selectors)

```
0x8da5cb5b [src] owner()
0x715018a6 [src] renounceOwnership()
0xf2fde38b [src] transferOwnership(address)
```

Facet 0x8dEbFF25EC2d6ABb3bF73F41E5B8b5a99917DFC0 (3 selectors)

```
0xee41f9d1 [src] batchSetSupportsInterface(bytes4[], bool[])
0x6660e282 [src] registerAllPXONInterfaces()
0x54e7796f [src] setSupportsInterface(bytes4, bool)
```

Facet 0x77cE2eB410C74bB81D36A9553434Caa985c6527a (3 selectors)

```
0x3aa6f0ee [src] getRedemptionRequest(uint256)
0x0fc6f448 [src] processRedemption(uint256)
0xeeb3c910 [src] requestRedemption(uint256)
```

Facet 0x69110e5f087f2910257A87dAFBb7a3f59b013b2A (3 selectors)

```
0x9588dd3a [src] cashReserveE18(bytes32)
0xa022fc78 [src] recordCashReserve(bytes32, uint256, bytes32)
0xfbab5f4c [src] totalReserveUSDE18(bytes32[])
```

Facet 0x66017098AE601bb49e47581b81CA000310968aac (3 selectors)

```
0x3b6520cb [src] executeUpgrade(bytes32)
0x8d3c1c49 [src] scheduleUpgrade(bytes, address, bytes, uint256)
0x5ba1c1a9 [4b] TIMELOCK_DELAY()
```

Facet 0xc57A4415C7e365712E546987def28B59082C127B (3 selectors)

0x5c60da1b [src] implementation()
0x3659cfe6 [src] upgradeTo(address)
0x4f1ef286 [src] upgradeToAndCall(address,bytes)

Facet 0x553DaCe32909730e1851Fe55331D178Ee79f655A (3 selectors)

0x900cf0cf [src] epoch()
0xb669ce0b [src] getRebaseMultiplier()
0x7a43e23f [src] rebase(uint256,int256)

Facet 0x54cC10A1AB6C4d1F9e925FF3967Ee90F09b95939 (2 selectors)

0x9dc29fac [src] burn(address,uint256)
0x40c10f19 [src] mint(address,uint256)

Facet 0xF93A8444bBfe59eBB4fCEc0c564d045a7D50daa6 (2 selectors)

0x03bca7d3 [src] getPriceFloor()
0xbd01bb32 [src] getTargetPrice()

Facet 0x8e253e21DdeE13aA365a11C72816eC0Bc75DB3FD (2 selectors)

0x0748ad14 [src] grantRoleWithRecord(bytes32,address)
0x3f8923c9 [src] revokeRoleWithRecord(bytes32,address)

Facet 0xba8dB0ad3db771224D25D3AFab22E04b95649667 (2 selectors)

0x47af267b [src] isFeatureEnabled(bytes32)
0xf2c4bc9e [src] setFeature(bytes32,bool)

Facet 0x1c8bF6224D4d70873D7401a3EbB2F6dA01762246 (2 selectors)

0xd7a39ed8 [src] evidenceSeq()
0x4f025f05 [src] recordEvidence(bytes32,bytes32,bytes)

Facet 0x8559c79b0D02f123c8945975beA8322cC9C5C27A (2 selectors)

0x1291f79d [src] rescueNative(address,uint256)
0xe5711e8b [src] rescueToken(address,address,uint256)

Facet 0xd09fD383f64A71c6f79409c2b3f85F4C23212961 (2 selectors)

0x5153d467 [src] isSupportedChain(uint256)
0x280edb06 [src] setSupportedChain(uint256,bool,bytes32)

Facet 0x5e725c33A2319339Bf9d290144f944F3Fd4CD4aD (2 selectors)

0x557bbf7f [src] isProofUsed(bytes32)
0x9bb74a80 [src] markProofUsed(bytes32)

Facet 0xeAa3602B855e8d06452f42F77a9D1F3d5486CEeb (2 selectors)

0x36eea402 [src] receiveCrossChain(uint256,bytes32,bytes,bytes)
0x0de84668 [src] sendCrossChain(uint256,bytes32,bytes)

Facet 0x203f453D7343DED6Ef035b6fCe4ca7943C0ee6BC (2 selectors)

0x47e7ef24 [src] deposit(address,uint256)
0xf3fef3a3 [src] withdraw(address,uint256)

Facet 0x7B40D027227b245ED9cF84B64E952b8d95797898 (2 selectors)

0x8272e138 [src] getAdapter(bytes32)
0x5cbc01f8 [src] setAdapter(bytes32,address,bool)

Facet 0xc4b7FbfF19F18237bD4E284ba2C921182e7b7f62 (2 selectors)

0xeeef37ab4 [src] getPegReference(address)
0x60c45a7d [src] setPegReference(address,bytes32,uint256)

Facet 0x08e382c26E65d29B25d3a1feB493C080a6A9d56C (1 selectors)

0x1f931c1c [4b] diamondCut((address,uint8,bytes4[])[],address,bytes)

Facet 0x0c3A9A9cAA6A7221394BE358F1003D4B101bCBd7 (1 selectors)

0x5123bacc [src] setKYC(address,bool)

Facet 0xF8075E26166a14E3412FBe24643732f7678dbc50 (1 selectors)

0x9394385d [src] setAISignal(bytes32,bytes32,uint256,bytes)

Facet 0xf53040B37f011Af12831fd74AD6Ee57a70343b31 (1 selectors)

0x66da0e2d [src] priceSources()

7. PXON2 Source Function Inventory

Every function declaration parsed from `Security_back_token/PXON2/*.sol`, grouped by file and contract. Legend: `[pub]`=public, `[ext]`=external, `[int]`=internal, `[prv]`=private, `[def]`=default. Mutability `[v]`=view `[p]`=pure `[$]`=payable. `&` = selector is currently routed by the live diamond; `%E` = declared in source but not exposed on-chain.

Files parsed: 59
Functions declared: 1020
public / external: 908
Routed on-chain: 567

Security_back_token/PXON2/01_CoreERC20.sol

contract CoreERC20FacetV4 (24 fn)

```
& 0x06fdde03 [ext] v name() !' (string memory)
& 0x95d89b41 [ext] v symbol() !' (string memory)
& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x18160ddd [ext] v totalSupply() !' (uint256)
& 0xd5abeb01 [ext] v maxSupply() !' (uint256)
& 0x70a08231 [ext] v balanceOf(address a) !' (uint256)
& 0xdd62ed3e [ext] v allowance(address o, address s) !' (uint256)
& 0x095ea7b3 [ext] approve(address spender, uint256 amount) !' (bool)
& 0x39509351 [ext] increaseAllowance(address spender, uint256 delta) !' (bool)
& 0xa457c2d7 [ext] decreaseAllowance(address spender, uint256 delta) !' (bool)
& 0xa9059cbb [ext] transfer(address to, uint256 amount) !' (bool)
& 0x23b872dd [ext] transferFrom(address from, address to, uint256 amount) !' (bool)
& 0x2535f762 [ext] transferWithData(address to, uint256 amount, bytes calldata) !' (bool)
& 0xee532f31 [ext] transferFromWithData(address from, address to, uint256 amount, bytes calldata) !' (bool)
& 0x6af0313d [ext] v pegEnforcer() !' (address)
& 0x7d6ba9d8 [ext] v pegKillSwitch() !' (bool)
& 0xd4bd8ba2 [ext] v shieldKillSwitch() !' (bool)
& 0xd4b53324 [ext] v killerKillSwitch() !' (bool)
& 0x68a394cf [ext] setPegEnforcer(address a)
& 0xfa9866a4 [ext] setPegKillSwitch(bool v)
& 0x1b8ae643 [ext] setShieldKillSwitch(bool v)
& 0x4cfd017e [ext] setKillerKillSwitch(bool v)
[int] _approve(address owner, address spender, uint256 amount)
[int] _transfer(address from, address to, uint256 amount)
```

contract PermitFacet (3 fn)

```
& 0x3644e515 [ext] v DOMAIN_SEPARATOR() !' (bytes32)
& 0x7ecebe00 [ext] v nonces(address owner) !' (uint256)
& 0xd505accf [ext] permit(address owner, address spender, uint256 value,
uint256 deadline, uint8 v, bytes32 r, bytes
32 s_)
```

contract InitERC20Facet (5 fn)

```
& 0xc380a285 [ext] initERC20(string calldata _name, string calldata _symbol, uint8 _dec)
& 0x7438b937 [ext] setDomainSeparator(bytes32 ds)
& 0xa4f29aad [ext] setTokenName(string calldata v)
& 0xba51b1b4 [ext] setTokenSymbol(string calldata v)
& 0xf2cf47be [ext] setTokenDecimals(uint8 v)
```

Security_back_token/PXON2/02_IssuanceCompliance.sol

contract MintBurnFacetProd (2 fn)

```
& 0x40c10f19 [ext] mint(address to, uint256 amount)
& 0x9dc29fac [ext] burn(address from, uint256 amount)
```

contract MintBurnFacet (2 fn)

```
& 0xbd01bb32 [ext] v getTargetPrice() !' (uint256)
& 0x03bca7d3 [ext] v getPriceFloor() !' (uint256)
```

contract ERC1400IssuanceFacet (8 fn)

```
& 0x2f1cae85 [ext] v isIssuable() !' (bool)
& 0x9ab309ca [ext] setIssuable(bool v)
& 0xbb3acde9 [ext] issue(address tokenHolder, uint256 value, bytes calldata data)
& 0xe77c646d [ext] redeem(uint256 value, bytes calldata data)
& 0x9675193c [ext] redeemFrom(address tokenHolder, uint256 value, bytes calldata data)
& 0x2535f762 [ext] transferWithData(address to, uint256 value, bytes calldata) !' (bool)
& 0xee532f31 [ext] transferFromWithData(address from, address to, uint256 value, bytes calldata) !' (bool)
& 0x36d948d1 [ext] v issuanceNonce(address holder) !' (uint256)
```

contract ERC1400ComplianceFacet (19 fn)

```
& 0x1badb25c [ext] v canTransfer(address to, uint256 value, bytes calldata) !' (bool, bytes1, bytes32)
& 0x122eb575 [ext] v canTransferFrom(address from, address to, uint256 value, bytes calldata) !' (bool, bytes1, bytes32)
& 0x9945e70e [ext] v canTransferByPartition(address from, address to, bytes32 partition, uint256 value, bytes calldata) !' (bool, bytes1, bytes32)
& 0x8eda1889 [ext] activateCompliance(bool erc1400, bool erc3643)
& 0x3643adb5 [ext] batchSetKYC(address[] calldata investors, bool[] calldata approved)
& 0x6e72a6bc [ext] setKYCApproved(address investor, bool v)
& 0x67315b3e [ext] setRiskLevel(address investor, uint8 level)
& 0xd01dd6d2 [ext] setBlacklisted(address investor, bool v)
& 0x9281aa0b [ext] setWhitelisted(address investor, bool v)
& 0x953715ae [ext] setTransferLimit(address investor, uint256 singleMax, uint256 dailyMax)
& 0xe7eeff76 [ext] setGlobalLimits(uint256 singleMax, uint256 dailyMax)
& 0xe8e5a085 [ext] setRedFlagged(address investor, bool v)
```



```

& 0xe77db232 [ext] v isKYCApproved(address investor) !' (bool)
& 0x1615f2d4 [ext] v getRiskLevel(address investor) !' (uint8)
& 0xfe575a87 [ext] v isBlacklisted(address investor) !' (bool)
& 0x3af32abf [ext] v isWhitelisted(address investor) !' (bool)
& 0xc14d1e4c [ext] v isRedFlagged(address investor) !' (bool)
& 0x5edf9e61 [ext] v getTransferLimit(address investor) !' (uint256 single, uint256 daily, uint256 dailyUsed)
& 0x5b090ed6 [ext] v complianceStatus() !' (bool erc1400Active, bool erc3643Active, bool paused)

```

Security_back_token/PXON2/03_IdentityRoles.sol

contract ERC3643IdentityFacet (18 fn)

```

& 0xb9209e33 [ext] v isVerified(address investor) !' (bool)
& 0xf0eb5e54 [ext] v identity(address investor) !' (address)
& 0x7e42683b [ext] v investorCountry(address investor) !' (uint16)
& 0xc3c5a547 [ext] v isRegistered(address investor) !' (bool)
& 0x5ad0c0bf [ext] v getIdentityRecord(address investor) !' (IdentityRecord memory)
& 0x05abe2ec [ext] v isCountryBlocked(uint16 country) !' (bool)
& 0xd7e64c00 [ext] v investorCount() !' (uint256)
& 0xc79339e2 [ext] v getComplianceModule() !' (address)
& 0x454a03e0 [ext] registerIdentity(address investor, address identityContract, uint16 country)
& 0x8e098ca1 [ext] updateIdentity(address investor, address identityContract)
& 0x3b239a7f [ext] updateCountry(address investor, uint16 country)
& 0xa8d29d1d [ext] deleteIdentity(address investor)
& 0x9c5db6d9 [ext] batchRegisterIdentities(address[] calldata investors,
    address[] calldata identities_,
    uint
    t16[] calldata countries)
& 0x8db7b007 [ext] setCountryBlocked(uint16 country, bool blocked)
& 0x35e896c1 [ext] setMaxInvestors(uint256 max)
& 0x423db9c7 [ext] setComplianceModule(address module)
& 0x1dabe830 [ext] setSequencerFeed(address , uint256)
& 0x3e54e07e [ext] p getSequencerFeed() !' (address feed, uint256 gracePeriod)

```

contract KYCAMLFacet (4 fn)

```

& 0xe77db232 [ext] v isKYCApproved(address investor) !' (bool)
& 0x1615f2d4 [ext] v getRiskLevel(address investor) !' (uint8)
& 0x5123bacc [ext] setKYC(address investor, bool approved)
& 0x67315b3e [ext] setRiskLevel(address investor, uint8 level)

```

contract RolesFacet (4 fn)

```

& 0x91d14854 [ext] v hasRole(bytes32 role, address account) !' (bool)
& 0x248a9ca3 [ext] p getRoleAdmin(bytes32) !' (bytes32)
& 0x2f2ff15d [ext] grantRole(bytes32 role, address account)
& 0xd547741f [ext] revokeRole(bytes32 role, address account)

```

contract RoleAdminActionsFacet (2 fn)

```

& 0x0748ad14 [ext] grantRoleWithRecord(bytes32 role, address account)
& 0x3f8923c9 [ext] revokeRoleWithRecord(bytes32 role, address account)

```

Security_back_token/PXON2/04_PartitionRedemption.sol

contract ERC1400PartitionFacet (9 fn)

```

& 0x740ab8f4 [ext] v partitionsOf(address) !' (bytes32[] memory parts)
& 0x30e82803 [ext] v balanceOfByPartition(bytes32 partition, address investor) !' (uint256)
& 0x69598efe [ext] v totalPartitions() !' (uint256)
& 0x6d77cad6 [ext] v isOperatorForPartition(bytes32 partition, address operator, address holder) !' (bool)
& 0xf3d490db [ext] transferByPartition(bytes32 partition, address to, uint256 value, bytes calldata data) !'
(bytes32)
& 0x8c0dee9c [ext] operatorTransferByPartition(bytes32 partition, address from, address to, uint256 value,
    bytes cal
    ldata data, bytes calldata opData) !' (bytes32)
& 0x103ef9e1 [ext] authorizeOperatorByPartition(bytes32 partition, address operator)
& 0x168ecec5 [ext] revokeOperatorByPartition(bytes32 partition, address operator)
& 0x069c6631 [ext] seedPartitionBalance(address investor, bytes32 partition, uint256 amount)

```

contract RedemptionFacet (3 fn)

```

& 0xeeb3c910 [ext] requestRedemption(uint256 tokenAmount)
& 0x0fc6f448 [ext] processRedemption(uint256 requestId)
& 0x3aa6f0ee [ext] v getRedemptionRequest(uint256 requestId) !' (RedemptionRequest memory)

```

contract RedemptionVaultFacet (8 fn)

```

& 0xf1fb0036 [ext] initRedeemVault(uint256 minCollateralBps, uint256 maxPerTx, uint256 maxPerDay)
& 0xf0920278 [ext] depositUSDW(uint256 amount)
& 0x288e5c58 [ext] withdrawUSDW(address to, uint256 amount)
& 0x1253b004 [ext] setMaxRedeemPerTx(uint256 v)
& 0xec143adc [ext] setMaxRedeemPerDay(uint256 v)
& 0xc2f07cc5 [ext] setMinCollateralBps(uint256 v)
& 0xa30c7bb0 [ext] v getRedeemRecord(bytes32 key) !' (uint256)
& 0x6ff4ab8a [ext] v vaultStats() !' (uint256 usdwBalance,
    uint256 redemptionReserve,
    uint256 liquidityReserve,

    uint256 freeBalance,
    uint256 maxPerTx,
    uint256 maxPerDay)

```

Security_back_token/PXON2/05_PoRNAV.sol

contract PorFacet (11 fn)

```

& 0x06c83f1c [ext] updateSeriesNAV(uint256 seriesId, uint256 newNAV)
& 0xf335d56a [ext] batchUpdateNAV(uint256[] calldata seriesIds, uint256[] calldata newNAVs)
& 0x30421927 [ext] v totalNAV() !' (uint256)
& 0xa291a842 [ext] v getNAVPerToken() !' (uint256)
& 0xe92fcab2 [ext] v getSeriesNAVPerToken(uint256 seriesId) !' (uint256)
& 0xedd129a9 [ext] setPorFeed(address feed, uint256 maxStaleness)
& 0x54ed0253 [ext] v getPorReserve() !' (uint256 reserveE18, uint256 updatedAt, bool isFresh)
& 0x90f48062 [ext] v getBackingRatioBps() !' (uint256)

```

```

& 0x18b9be2e [ext] v checkMintPolicy(uint256 mintAmount) !' (bool ok, string memory reason)
& 0x1a9ade51 [ext] v porFeedAddress() !' (address)
& 0x9b8417b8 [ext] v porMaxStaleness() !' (uint256)

```

contract PorSnapshotFacet (10 fn)

```

& 0xdeb8c641 [ext] v getSnapshotTimestamp() !' (uint256)
& 0xa1dae74b [ext] v porSnapshotHash() !' (bytes32)
& 0x534f8e98 [ext] v setPorSnapshotHash(bytes32 proofHash)
& 0x0e0b0db6 [ext] v getCertFeed() !' (address, uint256)
& 0x4aafbddb [ext] v isSnapshotFresh(uint256 maxAge) !' (bool)
& 0x795d5bb4 [ext] v setCertFeed(address feed)
& 0xc137ec56 [ext] v setCertFeedMaxStaleness(uint256 maxStaleness)
& 0x76f10ad0 [ext] v getSnapshot(uint256 index) !' (PORSnapshot memory)
& 0xcb22af4c [ext] v latestSnapshot() !' (PORSnapshot memory)
& 0x098ab6a1 [ext] v snapshotCount() !' (uint256)

```

contract PorPolicyFacet (7 fn)

```

& 0xdb249f10 [ext] v getMinReserveRatio() !' (uint256)
& 0x58a41a9c [ext] v isMintingDisabledOnReserveFailure() !' (bool)
& 0x83685135 [ext] v setMinReserveRatio(uint256 bps)
& 0xb68f3f02 [ext] v setMintingDisabledOnReserveFailure(bool disabled)
& 0x61458b4e [ext] v checkMintPolicy(uint256 mintAmount, uint256 currentSupply) !' (bool ok, string memory reason)
& 0x7ca7a971 [ext] v enforceMintPolicy(uint256 mintAmount, uint256 currentSupply)
& 0xc6dcc97c [ext] v getLastPolicyCheck() !' (uint256 timestamp, bool passed)

```

contract PoRMultiChainFacet (9 fn)

```

& 0x1483a234 [ext] v chainReserveUSDE18(uint256 chainId) !' (uint256)
& 0xec7c9607 [ext] v setChainReserveUSDE18(uint256 chainId, uint256 amount, bytes32 proofHash, uint64)
& 0x78c8f674 [ext] v getReserveSnap(uint256 chainId, address token) !' (uint256)
& 0x96bb9da5 [ext] v setReserveSnap(uint256 chainId, address token, uint256 amount, bytes32 proofHash, uint64)
& 0xf6be0afe [ext] v getReserveSnapSafe(uint256 chainId, address token) !' (uint256 amount, bool valid)
& 0x688595a0 [ext] v totalReserveUSDE18(uint256[] calldata chainIds) !' (uint256 total)
& 0x4c4bf9ea [ext] v totalReserveTokenPairsUSDE18(uint256[] calldata chainIds, address[] calldata tokens) !'
(uint256 total)
& 0x1d29777b [ext] v getMultiChainConfig() !' (uint256 chainCount, uint256 maxStaleness)
& 0xdc6e1022 [ext] v setMultiChainConfig(uint256 , uint256 maxStaleness)

```

Security_back_token/PXON2/06_SeriesDocController.sol

contract SeriesEngineFacet (15 fn)

```

& 0x948b5165 [ext] createSeries(string calldata _name,
    address _spv,
    uint256 _issuancePrice,
    uint256
    _navValue,
    uint256 _minInvestment,
    uint256 _maxInvestment,
    uint256 _lockPeriod,
    bytes32 _assetHas
    h,
    bytes32 _docRoot) !' (uint256 id)
& 0xb0a9b406 [ext] v setSeriesFrozen(uint256 id, bool frozen)
& 0xeac1ee47 [ext] v exitSeries(uint256 id)
& 0xb50614a8 [ext] v updateSeriesDocRoot(uint256 id, bytes32 assetHash, bytes32 docRoot)
& 0x1160da15 [ext] v getSeries(uint256 id) !' (SeriesMeta memory)
& 0xd7f2c0ef [ext] v seriesCount() !' (uint256)
& 0xf5ceb211 [ext] v getSeriesBalance(address investor, uint256 seriesId) !' (uint256)
& 0x4510825f [ext] v getSeriesInvestors(uint256 id) !' (address[] memory)
& 0x9bccdf41 [ext] v buySeriesTokens(uint256 seriesId, uint256 usdwAmount)
& 0x14b9abc9 [ext] v sellSeriesTokens(uint256 seriesId, uint256 tokenAmount)
& 0x75deb2a7 [ext] v fulfillRedemption(uint256 requestId)
& 0x9219e510 [ext] v cancelRedemption(uint256 requestId)
& 0x7f407edf [ext] v setSpread(uint256 buyBps, uint256 sellBps)
& 0x3aa6f0ee [ext] v getRedemptionRequest(uint256 id) !' (RedemptionRequest memory)
& 0xf7d57f1c [ext] v getInvestorRedemptions(address investor) !' (uint256[] memory)

```

contract ERC1400DocumentFacet (10 fn)

```

& 0xb10d6b41 [ext] v getDocument(bytes32 docName) !' (string memory uri, bytes32 hash, uint256 updatedAt)
& 0x9fa5f50b [ext] v getAllDocuments() !' (bytes32[] memory)
& 0xcbfe6d41 [ext] v documentExists(bytes32 docName) !' (bool)
& 0x010648ca [ext] v setDocument(bytes32 docName, string calldata uri, bytes32 contentHash)
& 0xc3501848 [ext] v removeDocument(bytes32 docName)
& 0xba69cb9b [ext] v batchSetDocuments(bytes32[] calldata names, string[] calldata uris, bytes32[] calldata hashes)
& 0xd7b84c9b [ext] v setSeriesDocument(uint256 seriesId, bytes32 key, string calldata uri, bytes32 hash)
& 0xf4e0c809 [ext] v getSeriesDocument(uint256 seriesId, bytes32 key) !' (string memory uri, bytes32 hash, uint256
ts)
& 0x15a4923c [ext] bootstrapDocuments(string calldata auditUri, bytes32 auditHash,
    string calldata porUri, byt
es32 porHash,
    string calldata wpUri, bytes32 wpHash,
    string calldata isoUri, bytes32 isoHash,
    str
ing calldata legalUri, bytes32 legalHash)
[int] _setDoc(AppStorage storage s, bytes32 key, string memory uri, bytes32 hash)

```

contract ERC1400ControllerFacet (8 fn)

```

& 0x4c783bf5 [ext] v isControllable() !' (bool)
& 0xf77c4791 [ext] v controller() !' (address)
& 0x92eefe9b [ext] v setController(address newCtrl)
& 0xf282527a [ext] v controllerTransfer(address from, address to, uint256 amount, bytes calldata data, bytes
calldata)
& 0x2bc6acc3 [ext] v controllerRedeem(address from, uint256 amount, bytes calldata data, bytes calldata)
& 0x4f4bad4e [ext] v getControllerAction(uint256 index) !' (ControllerAction memory)
& 0xe02dbc5 [ext] v controllerActionCount() !' (uint256)
[int] _logAction(AppStorage storage s, address from, address to, uint256 amount, uint256 seriesId,
string memory
reason, bool isRedeem)

```

Security_back_token/PXON2/07_OracleFeeds.sol

contract PriceFeedFacetBSC (14 fn)

```
& 0x9358928b [ext] v circulatingSupply() !' (uint256)
& 0x52d3f41a [ext] v marketCapUSD() !' (uint256)
& 0xf915e6d [ext] v reserveBacking() !' (uint256 reserveE18, uint256 ratioBps)
& 0xb8f01c3b [ext] v liquidityUSD() !' (uint256)
& 0x9a105d2b [ext] p pegTargetE18() !' (uint256)
& 0x6301a747 [ext] p priceFeedVersion() !' (string memory)
& 0xb86e7822 [ext] v feedSummary() !' (uint256 supply, uint256 totalNAV, uint256 navPerToken,
uint256 usdwReserve, uint
256 porReserveE18, bool porFresh)
& 0x5853b613 [ext] touchFeed()
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt, uint256 updatedAt,
uint80 answered
InRound)
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
& 0x7284e416 [ext] p description() !' (string memory)
& 0x54fd4d50 [ext] p version() !' (uint256)
& 0x93a408d7 [ext] p priceUSD() !' (uint256)
& 0xd7355955 [ext] p isPegged() !' (bool)
```

contract CanonicalOracleFacet (14 fn)

```
& 0x7542f23a [ext] v pegPrice() !' (uint256)
& 0x18ff3c8b [ext] pokePeg()
& 0xb95ed06f [ext] v priceOf(address) !' (uint256)
& 0xe60211dc [ext] setQuorum(uint16 quorum)
& 0x63528565 [ext] setSource(uint8 slot, address source, string calldata label)
& 0x7df153d1 [ext] v source(uint8 slot) !' (address)
& 0xb893fe85 [ext] v sourceLabel(uint8 slot) !' (string memory)
& 0xf914f712 [ext] v usdwLatestAnswer() !' (int256)
& 0x30dd6a6d [ext] v usdwLatestRound() !' (uint256)
& 0xae87a07 [ext] v usdwLatestTimestamp() !' (uint256)
& 0xc1bc8ad9 [ext] p usdwOracleDecimals() !' (uint8)
& 0xe5c50828 [ext] p usdwOracleDescription() !' (string memory)
& 0x0ffc1a32 [ext] p usdwOracleVersion() !' (uint256)
& 0xf3e316d4 [ext] v usdwLatestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt, uint256 updatedAt,
uint80 answe
redInRound)
```

contract OracleViewFacet (12 fn)

```
& 0x833b1fce [ext] v getOracle() !' (address)
& 0x98d5fdca [ext] v getPrice() !' (uint256)
& 0xae8de588 [ext] v getMaxStaleness() !' (uint256)
& 0x100bc9e1 [ext] setMaxStaleness(uint256 v)
& 0x335e077a [ext] refreshAndVerify()
& 0x7adb9f73 [ext] setOracle(address oracle)
& 0x63929d58 [ext] v CHAINLINK_PEG_ANSWER() !' (int256)
& 0x8ba83c0b [ext] p PEG_USD() !' (uint256)
& 0x3f903940 [ext] v feedMeta() !' (address feed, uint256 staleness, uint256 lastRefresh)
& 0x16da057b [ext] v getLastVerified() !' (uint256)
& 0xdd47efb7 [ext] p REQUIRED_DECIMALS() !' (uint8)
& 0xacbbca7b [ext] p REQUIRED_DESCRIPTION() !' (string memory)
```

contract PriceLogicFacet (6 fn)

```
& 0xc97ab421 [ext] addPriceSource(address source)
& 0x623b6549 [ext] removePriceSource(address source)
& 0x78bc2ff9 [ext] v isPriceSource(address source) !' (bool)
& 0xbb2ec9af [ext] v getPriceSources() !' (address[] memory)
& 0x8d6cc56d [ext] updatePrice(uint256 newPrice)
& 0xb7f88320 [ext] v getLastUpdateTime() !' (uint256)
```

Security_back_token/PXON2/08_Security.sol

contract EmergencyFacet (8 fn)

```
& 0xb1e3268b [ext] emergencyStop(string calldata reason)
& 0xb187bd26 [ext] v isPaused() !' (bool)
& 0x7faf6a79 [ext] v lastPause() !' (uint256)
& 0x39c09ef5 [ext] v lastUnpause() !' (uint256)
& 0x60732533 [ext] pause(uint8 level, string calldata reason)
& 0xcd11d1d2 [ext] pauseEx(uint8 level, string calldata reason)
& 0xb6d9fa09 [ext] unpause(uint8 , string calldata reason)
& 0x8893172e [ext] unpauseEx(uint8 , string calldata reason)
```

contract ComplianceFacet (7 fn)

```
& 0xfe575a87 [ext] v isBlacklisted(address account) !' (bool)
& 0x8456cb59 [ext] pause()
& 0x3f4ba83a [ext] unpause()
& 0x5c975abb [ext] v paused() !' (bool)
& 0xd01dd6d2 [ext] setBlacklisted(address account, bool v)
& 0x9281aa0b [ext] setWhitelisted(address account, bool v)
& 0xc4590d3f [ext] setLimits(uint256 singleMax, uint256 dailyMax)
```

contract PegPolicyFacet (5 fn)

```
& 0x62e0dbde [ext] v assertPegE18(uint256 priceE18)
& 0x3e69fa76 [ext] initPegPolicyDefaults()
& 0x0f64d73e [ext] v isPegHealthyE18(uint256 priceE18) !' (bool)
& 0x7cf981f2 [ext] v pegMinE18() !' (uint256)
& 0x4477324b [ext] setPegPolicyE18(uint256 minE18, uint256 maxE18)
```

contract UniversalPegGuard (7 fn)

```
& 0x8e40b35e [ext] v enforceInvariant(address , address , uint256)
& 0x2cedb376 [ext] v getGuardConfig() !' (bool active, uint256 maxBandBps)
& 0x77f62e04 [ext] v initUniversalGuard(address , address , uint256 minE18, uint256
maxBandBps)
& 0x48334596 [ext] v isAMMAddress(address pool) !' (bool)
& 0x3ea6e071 [ext] setExemption(address account, bool exempt)
```

```

& 0x2e3cfe49 [ext] setGuardPaused(bool v)
& 0x703b61cb [ext] setMaxBandBps(uint256 bps)

```

contract ISOControlFacet (17 fn)

```

& 0x99737793 [ext] v continuityState() !' (uint8)
& 0x5d6dfa9e [ext] setContinuityMode(uint8 mode, bytes32 ref)
& 0x521fcd32 [ext] setFunctionPaused(bytes4 selector, bool paused_, bytes32 ref)
& 0xf7fb3e42 [ext] setGlobalHalt(bool halted, bytes32 ref)
& 0x7814103d [ext] setRedFlag(address account, bool flagged, bytes32 ref)
& 0x78e785ad [ext] v isRedFlag(address account) !' (bool)
& 0x71020dff [ext] v checkRedFlagGuard(bytes4 selector, address from, address to) !' (bool blocked)
& 0xf50bf174 [ext] registerObligation(bytes32 id, bytes32 ref)
& 0x7f9e0fd2 [ext] v getObligation(bytes32 id) !' (bytes32)
& 0x6147c173 [ext] anchorEvidence(bytes32 id, bytes32 hash_)
& 0x4a7221a0 [ext] v getEvidence(bytes32 id) !' (bytes32)
& 0x67febe72 [ext] setSignerApproved(address signer, bool approved)
& 0xc0b4a4def [ext] v signerApproved(address signer) !' (bool)
& 0x87325830 [ext] setProviderApproved(bytes32 id, bool approved)
& 0x4b9f9fbf [ext] v providerApproved(bytes32 id) !' (bool)
& 0x2eb99313 [ext] emitWebhook(bytes32 topic, bytes calldata data)
& 0x6d9b99eb [ext] v webhookNonce(bytes32) !' (uint256)

```

Security_back_token/PXON2/09_Plugins.sol

contract StakingPluginFacet (9 fn)

```

& 0xe29a9370 [ext] v isStakingEnabled() !' (bool)
& 0x16765391 [ext] v stakedBalanceOf(address account) !' (uint256)
& 0x31d7a262 [ext] v pendingRewards(address account) !' (uint256)
& 0xa694fc3a [ext] stake(uint256 amount)
& 0x2e17de78 [ext] unstake(uint256 amount)
& 0x372500ab [ext] claimRewards()
& 0x9e447fc6 [ext] setRewardRate(uint256 bps)
& 0xbce8567e [ext] setStakingEnabled(bool v)
[internal] _accrueRewards(AppStorage storage s, address account)

```

contract LockingFacet (4 fn)

```

& 0x929ec537 [ext] v getLockedAmount(address account) !' (uint256)
& 0xdbed1e88 [ext] v getUnlockTime(address account) !' (uint256)
& 0xe2ab691d [ext] lock(address account, uint256 amount, uint256 unlockAt)
& 0x2f6c493c [ext] unlock(address account)

```

contract VestingPluginFacet (4 fn)

```

& 0x5e60bf7b [ext] createVesting(address beneficiary, uint256 totalAmount,
uint64 startTime, uint64 cliffDuration, u
int64 totalDuration)
& 0xe4bf01a8 [ext] v releasable(uint256 id) !' (uint256)
& 0x37bdc99b [ext] release(uint256 id)
& 0x2f5c4fe1 [ext] setVestingEnabled(bool v)

```

contract GovernanceFacet (4 fn)

```

& 0x5c19a95c [ext] delegate(address delegatee)
& 0x587cde1e [ext] v delegates(address account) !' (address)
& 0x8d22ea2a [ext] v delegateOf(address account) !' (address)
& 0x9ab24eb0 [ext] v getVotes(address account) !' (uint256)

```

contract FeatureFlagsFacet (2 fn)

```

& 0x47af267b [ext] v isFeatureEnabled(bytes32 key) !' (bool)
& 0xf2c4bc9e [ext] setFeature(bytes32 key, bool enabled)

```

contract ConfigFacet (4 fn)

```

& 0x6dd5b69d [ext] v getConfig(bytes32 key) !' (uint256)
& 0x21f8a721 [ext] v getAddress(bytes32 key) !' (address)
& 0x15fe96dc [ext] setConfig(bytes32 key, uint256 value)
& 0xca446dd9 [ext] setAddress(bytes32 key, address value)

```

contract AuditBoardFacet (2 fn)

```

& 0xd7a39ed8 [ext] v evidenceSeq() !' (uint256)
& 0x4f025f05 [ext] recordEvidence(bytes32 evidenceHash, bytes32 contextHash, bytes calldata metadata)

```

contract AdminUtilsFacet (2 fn)

```

& 0xe5711e8b [ext] rescueToken(address token, address to, uint256 amount)
& 0x1291f79d [ext] rescueNative(address payable to, uint256 amount)

```

contract FXRatesFacet (4 fn)

```

& 0x630ed513 [ext] v getFXRate(bytes32 currency) !' (uint256 rateE18, uint64 updatedAt)
& 0x9871aa6e [ext] v quoteCurrencyToUSDWE18(uint256 amount, bytes32 currency) !' (uint256)
& 0x0a503bab [ext] v quoteUSDWToCurrencyE18(uint256 usdwAmount, bytes32 currency) !' (uint256)
& 0x53a24645 [ext] setFXRate(bytes32 currency, uint256 rateE18, uint64 timestamp)

```

contract MultiCurrencyReserveFacet (3 fn)

```

& 0x9588dd3a [ext] v cashReserveE18(bytes32 currency) !' (uint256)
& 0xa022fc78 [ext] recordCashReserve(bytes32 currency, uint256 amountE18, bytes32 proofHash)
& 0xfbab5f4c [ext] v totalReserveUSDE18(bytes32[] calldata currencies) !' (uint256 total)

```

contract GaslessFacet (11 fn)

```

& 0xa530d833 [ext] v gaslessNonce(address account) !' (uint256)
& 0x963a959e [ext] v isGaslessEnabled() !' (bool)
& 0x1b2d6577 [ext] v maxGasPerOp() !' (uint256)
& 0xf47c5cdb [ext] v isApprovedTarget(address target) !' (bool)
& 0x52636ebe [ext] v isOpExecuted(bytes32 opHash) !' (bool)
& 0xae52c569 [ext] v getApprovedTargets() !' (address[] memory)
& 0x395c3226 [ext] setGasless(bool enabled)
& 0x3a140e38 [ext] setMaxGasPerOp(uint256 gas_)
& 0xb7f9a748 [ext] setApprovedTarget(address target, bool approved)
& 0xc0890814 [ext] batchApproveTargets(address[] calldata targets, bool[] calldata approved)
& 0xbb44d15c [ext] executeGasless(address from, address to, bytes calldata data,
uint256 nonce, uint256 deadline, ui
nt256
uint8 v, bytes32 r, bytes32 s_) !' (bool, bytes memory)

```


Security_back_token/PXON2/10_Identity_Meta_Chain.sol

contract TokenIdentityFacet (10 fn)

```
& 0x7b96b8e0 [ext] v getTokenIdentity() !' (string memory name_, string memory symbol_, uint8 decimals_,
    string memory
    website, string memory logoURI, string memory cgId, string memory cmcSlug_)
& 0x995d9ab7 [ext] p getTags() !' (string[] memory tags)
& 0x0a8f41b4 [ext] v getSocialLinks() !' (string memory website)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
& 0xeb6db8f5 [ext] addChainDeployment(uint256 chainId, string calldata chainName, address contractAddr, uint32
    lzId)
& 0xd7f24abd [ext] v getChainDeployment(uint256 chainId) !' (ChainDeployment memory)
& 0xd25d23bd [ext] v getChainCount() !' (uint256 count)
& 0xae8a51ab [ext] initTokenIdentity(string calldata website, address resolver)
& 0x0570df66 [ext] setTokenLogo(string calldata uri)
& 0x291526ac [ext] v tokenLogo() !' (string memory)
```

contract CrossChainRegistryFacet (2 fn)

```
& 0x5153d467 [ext] v isSupportedChain(uint256 chainId) !' (bool)
& 0x280eb0b6 [ext] setSupportedChain(uint256 chainId, bool supported, bytes32 ref)
```

contract MarketDataFacet (4 fn)

```
& 0x4d96264c [ext] v getMarketData() !' (uint256 price, uint256 supply, uint256 totalNAV_, uint256 navPerToken)
& 0xcd238f76 [ext] v getMarketDataFull() !' (uint256 price, uint256 supply, uint256 totalNAV_, uint256 navPerToken,
    uint
    t256 porReserveE18, bool porFresh, uint256 backingRatioBps, uint256 usdwReserve)
& 0x597befa8 [ext] v priceUsdFloat() !' (uint256)
& 0x2f4a83e7 [ext] p marketDataVersion() !' (string memory)
```

contract UpgradeTimelockFacet (2 fn)

```
& 0x8d3c1c49 [ext] scheduleUpgrade(bytes calldata cutCalldata,
    address initAddress,
    bytes calldata initCalld
    ata,
    uint256 delay) !' (bytes32 id)
& 0x3b6520cb [ext] executeUpgrade(bytes32 id)
```

contract RebasePluginFacet (4 fn)

```
& 0x7c2d5327 [ext] v isRebaseEnabled() !' (bool)
& 0x4162640b [ext] v indexE18() !' (uint256)
& 0xca145b53 [ext] setIndexE18(uint256 index)
& 0x4b56de38 [ext] setRebaseEnabled(bool v)
```

contract AuthenticityGuardFacet (2 fn)

```
& 0x557bbf7f [ext] v isProofUsed(bytes32 proof) !' (bool)
& 0x9bb74a80 [ext] markProofUsed(bytes32 proof)
```

contract CatalogComplianceFacet (5 fn)

```
& 0xe0cdb2d9 [ext] v categoryEnabled(uint8 cat) !' (bool)
& 0x02c5d006 [ext] v categoryMask() !' (uint256)
& 0xf961d8b7 [ext] v functionAllowed(bytes4 selector) !' (bool)
& 0x4e29fe09 [ext] setCategoryEnabled(uint8 cat, bool enabled)
& 0xa96150df [ext] setSelectorPolicy(bytes4 selector, bool allowed)
```

Security_back_token/PXON2/11_ExtendedPlugins.sol

contract GovernorAdapterFacet (7 fn)

```
& 0x0c340a24 [ext] v governor() !' (address)
& 0xc42cf535 [ext] setGovernor(address _governor)
& 0x37558af5 [ext] propose(bytes calldata proposalData) !' (uint256 proposalId)
& 0xddf0b009 [ext] queue(uint256 proposalId)
& 0xfe0d94c1 [ext] execute(uint256 proposalId)
& 0x40e58ee5 [ext] cancel(uint256 proposalId)
& 0x943e8216 [ext] vote(uint256 proposalId, uint8 support)
```

contract BridgePluginFacet (4 fn)

```
& 0x69c366fe [ext] burnForBridge(uint256 amount, uint256 toChainId, bytes32 recipient)
& 0x495778a4 [ext] mintFromBridge(BridgeMintRequest calldata req, bytes calldata)
& 0x0c700d71 [ext] setBridgeEnabled(bool enabled)
& 0x20295abe [ext] setBridgeValidator(address validator, bool approved)
```

contract CrossChainAdapterFacet (2 fn)

```
& 0x0de84668 [ext] sendCrossChain(uint256 toChainId, bytes32 recipient, bytes calldata payload)
& 0x36eea402 [ext] receiveCrossChain(uint256 fromChainId, bytes32 txHash, bytes calldata payload, bytes calldata)
```

contract BridgeCompatFacet (2 fn)

```
& 0x47e7ef24 [ext] deposit(address token, uint256 amount)
& 0xf3fef3a3 [ext] withdraw(address token, uint256 amount)
```

contract AdapterRegistryFacet (4 fn)

```
& 0x9d2a954d [ext] v getAdapter(bytes32 protocol, bytes32 key) !' (address)
& 0xc5d16c6c [ext] v getHook(bytes32 protocol, bytes32 key) !' (bytes32)
& 0x7b34de77 [ext] setAdapter(bytes32 protocol, bytes32 key, address adapter)
& 0x0face053 [ext] setHook(bytes32 protocol, bytes32 key, bytes32 hookRef)
```

contract GatewayRegistryFacet (2 fn)

```
& 0x8272e138 [ext] v getAdapter(bytes32 id) !' (address adapter, bool enabled)
& 0x5cbc01f8 [ext] setAdapter(bytes32 id, address adapter, bool enabled)
```

contract VoucherFacet (8 fn)

```
& 0x9d11c3ac [ext] createVoucher(uint256 value, uint256 expiry, bytes32 voucherKey)
& 0x827ba03a [ext] claimVoucher(bytes32 voucherKey)
& 0x6f486781 [ext] sweepExpiredVoucher(bytes32 voucherKey)
& 0xf24f50bc [ext] v getDefaultVoucherExpiry() !' (uint256)
& 0x1a96e8bc [ext] v getVouchersByCreator(address creator) !' (bytes32[] memory)
& 0xf175075a [ext] v isVoucherActive(bytes32 voucherKey) !' (bool)
& 0x29ad0d05 [ext] v totalVoucherValue() !' (uint256)
& 0x439f30a4 [ext] v totalVouchersCreated() !' (uint256)
```

contract UpgradeabilityCompatFacet (3 fn)

```
& 0x5c60da1b [ext] v implementation() !' (address)
& 0x3659cfe6 [ext] p upgradeTo(address)
& 0x4f1ef286 [ext] p upgradeToAndCall(address , bytes calldata)
```

contract PegReferenceFacet (2 fn)

```
& 0xeeef37ab4 [ext] v getPegReference(address token) !' (bytes32 feedKey, uint256 tolerance)
& 0x60c45a7d [ext] v setPegReference(address token, bytes32 feedKey, uint256 tolerance)
```

contract AIOracleFacet (1 fn)

```
& 0x9394385d [ext] v setAISignal(bytes32 signalId, bytes32 dataHash, uint256 confidence, bytes calldata)
```

contract PriceSourcesFacet (1 fn)

```
& 0x66da0e2d [ext] v priceSources() !' (address[] memory sources)
```

contract RebaseElasticFacet (3 fn)

```
& 0x900cf0cf [ext] v epoch() !' (uint256)
& 0xb669ce0b [ext] v getRebaseMultiplier() !' (uint256)
& 0x74a3e23f [ext] v rebase(uint256 newEpoch, int256 supplyDelta)
```

contract RealDataFacet (14 fn)

```
& 0x80e0704b [ext] v cmcSlug() !' (string memory)
& 0xc2fe0fd8 [ext] v coingeckoId() !' (string memory)
& 0x3defeb04 [ext] v tokenWebsite() !' (string memory)
& 0x8820a08e [ext] v tokenLogoURI() !' (string memory)
& 0x6c02a931 [ext] v tokenName() !' (string memory)
& 0x7b61c320 [ext] v tokenSymbol() !' (string memory)
& 0xf968dfdd [ext] v setCMCSlug(string calldata slug)
& 0x4a2ff7d73 [ext] v setCoingeckoId(string calldata id)
& 0xc30ff6df [ext] v setLogoURI(string calldata uri)
& 0xb745ae93 [ext] v getRealMetrics() !' (RealMetrics memory m)
& 0x4ad17cdc [ext] v getPriceHistory(uint16 count) !' (uint256[] memory prices, uint256[] memory timestamps)
& 0x12d0b57c [ext] v recordSwap(address pair, address token, uint256 amount, bool isBuy)
& 0xa126f13a [ext] v snapshotHourly()
& 0x7406f76b [ext] v initRealData(address , address , string calldata website, string
calldata cmcS
lug_, string calldata cgId)
```

Security_back_token/PXON2/AppStorage.sol

library LibAppStorage (1 fn)

```
%E 0x9459e9ce [int] p diamondStorage() !' (AppStorage storage ds)
```

Security_back_token/PXON2/CG&CMC/contracts/IUSDWExternals.sol

interface AggregatorV3Interface (5 fn)

```
& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x7284e416 [ext] v description() !' (string memory)
& 0x54fd4d50 [ext] v version() !' (uint256)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId,
int256 answer,
uint256 startedAt
'
uint256 updatedAt,
uint80 answeredInRound)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId,
int256 answer,
uint256 startedAt,
uint256 updatedAt,
uint80 answeredInRound)
```

interface AggregatorV2Interface (5 fn)

```
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%E 0x668a0f02 [ext] v latestRound() !' (uint256)
%E 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (int256)
%E 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)
```

interface ICoreERC20Facet (4 fn)

```
& 0x18160ddd [ext] v totalSupply() !' (uint256)
& 0x06fdde03 [ext] v name() !' (string memory)
& 0x95d89b41 [ext] v symbol() !' (string memory)
& 0x313ce567 [ext] v decimals() !' (uint8)
```

interface IPriceFeedFacet (2 fn)

```
& 0x93a408d7 [ext] v priceUSD() !' (uint256)
%E 0xffb1a06d [ext] v priceIsStale() !' (bool)
```

interface IPorFacet (1 fn)

```
%E 0x0902f1ac [ext] v getReserves() !' (uint256)
```

interface IPoRMultiChainFacet (1 fn)

```
& 0x688595a0 [ext] v totalReserveUSDE18(uint256[] calldata chainIds) !' (uint256)
```

interface IPorSnapshotFacet (3 fn)

```
& 0x4aafbddb [ext] v isSnapshotFresh(uint256 maxAge) !' (bool)
& 0xdeb8c641 [ext] v getSnapshotTimestamp() !' (uint256)
& 0xa1dae74b [ext] v porSnapshotHash() !' (bytes32)
```

interface IUSDWMetaFacet (1 fn)

```
%E 0x67c859ab [ext] v getPoRMaxStaleness() !' (uint256)
```

interface ITokenIdentityFacet (2 fn)

```
& 0x0570df66 [ext] v setTokenLogo(string calldata logoURI)
& 0x291526ac [ext] v tokenLogo() !' (string memory)
```

interface IAutoListFacet (1 fn)

```
%E 0xaa16dda9 [ext] broadcastDiscovery(string calldata name,  
    string calldata website,  
    string calldata logo)
```

interface IOwnershipFacet (1 fn)

```
& 0x8da5cb5b [ext] v owner() !' (address)
```

Security_back_token/PXON2/CG&CMC/contracts/IUSDWMonetarySemantics.sol

interface IUSDWMonetarySemantics (26 fn)

```
%E 0xba4771ba [ext] p assetRole() !' (bytes32)  
%E 0x86c9de05 [ext] p assetClass() !' (bytes32)  
%E 0xb0c48a4b [ext] p marketParticipation() !' (bool)  
%E 0xf4fc7f65 [ext] p upgradeabilityPurpose() !' (bytes32)  
%E 0xe8945b66 [ext] p mintModel() !' (bytes32)  
%E 0x9acf4c1a [ext] p mintTrigger() !' (bytes32)  
%E 0x5c0ae241 [ext] p elasticity() !' (bool)  
%E 0x84550aa2 [ext] p tradableOnAMM() !' (bool)  
%E 0xdd26b4ed [ext] p sellExpected() !' (bool)  
%E 0x9c73ab30 [ext] p transferRole() !' (bytes32)  
%E 0x5c46b8e7 [ext] p liquidityModel() !' (bytes32)  
%E 0xd77be17a [ext] p liquiditySource() !' (bytes32)  
%E 0xfd236ad3 [ext] p dexLiquidityAuthoritative() !' (bool)  
%E 0x4a22ad4f [ext] p liquidityType() !' (bytes32)  
%E 0x13cf2b06 [ext] p liquidityProof() !' (bytes32)  
%E 0x3a0e11c8 [ext] p liquidityAuthoritativeSource() !' (bytes32)  
%E 0x33da7e26 [ext] p liquidityFakeable() !' (bool)  
%E 0xf7997916 [ext] p marketLiquidityRelevant() !' (bool)  
%E 0xf0464826 [ext] p realLiquidityFormula() !' (string memory)  
%E 0x35a8ea5d [ext] p emergencyControlModel() !' (bytes32)  
%E 0x04caa6dd [ext] p pauseReason() !' (bytes32)  
%E 0xc3ee477a [ext] p freezeConditions() !' (bytes32)  
%E 0x241e51f8 [ext] p seizure() !' (bool)  
%E 0x3f99e657 [ext] p monetarySemanticsBundle() !' (bytes32 assetRole_,  
    bytes32 assetClass_,  
    bytes32 mintModel_,  
  
    bytes32 transferRole_,  
    bytes32 liquidityModel_,  
    bytes32 emergencyControlModel_,  
    bool tradableOnAMM_,  
    bool dexLiquidityAuthoritative_,  
    bool seizure_)  
%E 0x0be245be [ext] p liquidityProtocolBundle() !' (bytes32 liquidityType_,  
    bytes32 liquidityModel_,  
    bytes32 liquiditySource_,  
    bytes32 liquidityProof_,  
    bytes32 liquidityAuthoritativeSource_,  
    bool liquidityFakeable_,  
  
    bool marketLiquidityRelevant_,  
    bool dexLiquidityAuthoritative_)  
%E 0x7f4209cb [ext] p monetarySemanticsVersion() !' (string memory)
```

Security_back_token/PXON2/CG&CMC/contracts/LibUSDWAggregatorStorage.sol

library LibUSDWAggregatorStorage (1 fn)

```
%E 0x86b714e2 [int] p s() !' (LogoStorage storage st)
```

Security_back_token/PXON2/CG&CMC/contracts/USDWCGCMCFeedAggregator.sol

contract USDWCGCMCFeedAggregator (23 fn)

```
%E 0x668a0f02 [ext] v latestRound() !' (uint256)  
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId,  
    int256 answer,  
    uint256 startedAt,  
  
    uint256 updatedAt,  
    uint80 answeredInRound)  
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId,  
    int256 answer,  
    uint256 startedAt,  
  
    uint256 updatedAt,  
    uint80 answeredInRound)  
& 0x50d25bcd [ext] v latestAnswer() !' (int256)  
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)  
%E 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (int256)  
%E 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)  
%E 0x6406f9db [ext] v getCanonicalSnapshot() !' (CanonicalSnapshot memory snap)  
%E 0x22bc2e8a [ext] v updateFeedRound() !' (uint80 roundId, int256 answer)  
    [int] v _compute() !' (int256 finalPrice8,  
    uint256 porPrice8,  
    uint256 oraclePrice8,  
  
    uint256 coverageBps,  
    uint256 reserveE18,  
    uint256 supplyE18,  
    uint8 status)  
    [int] v _readMultiChainReserve() !' (uint256)  
    [int] v _readOraclePrice8() !' (uint256 normalized8)  
    [int] v _updateVariance(uint256 porPrice8, uint256 oraclePrice8)  
%E 0xcc6e478e [ext] setRiskParameters(uint256 maxOracleDeviationBps_,  
    uint256 maxPremiumBps_)
```

```

uint256 porStal
nessOverride_,
uint256 minOracleVarianceBps_,
uint256 minPorVarianceBps_,
uint8 cgIterations_)
%E 0x748747e6 [ext] setKeeper(address keeper_)
%E 0x75829def [ext] transferAdmin(address newAdmin)
%E 0x16c38b3c [ext] setPaused(bool p)
%E 0x2bde8d2a [ext] setMinRefreshInterval(uint256 s_)
%E 0xf97cb290 [ext] setPorChainIds(uint256[] calldata ids)
%E 0x90c3f38f [ext] setDescription(string calldata d)
%E 0x6466afbe [ext] v getPorChainIds() !' (uint256[] memory)
[prv] p _maxU(uint256 a, uint256 b) !' (uint256)
[prv] p _sq(uint256 x) !' (uint256)

```

Security_back_token/PXON2/CG&CMC/contracts/USDWLogoFromReserveFacet.sol

contract USDWLogoFromReserveFacet (17 fn)

```

%E 0x40753e58 [ext] initializeLogoFacet(uint256[] calldata porChainIds_)
%E 0x2e6057ca [ext] setLogoURIs(string calldata excellent_,
string calldata healthy_,
string calldata stable_
,
string calldata warning_,
string calldata stale_)
%E 0x8424e581 [ext] setLogoIdentity(string calldata name_, string calldata website_)
%E 0xadba0b61 [ext] setLogoKeeper(address keeper_)
%E 0x7977e0a5 [ext] transferLogoAdmin(address newAdmin)
%E 0xa3cffa8b [ext] setLogoPorChainIds(uint256[] calldata ids)
%E 0x896f51cf [ext] setLogoStalenessOverride(uint256 maxAge)
%E 0xfea5272d [ext] syncLogoFromReserve() !' (uint8 newState, string memory newLogo)
%E 0x64f7b224 [ext] v getCurrentLogoState() !' (uint8 state, string memory logoURI, uint256 coverageBps, bool
porFresh)
%E 0x1c1be78c [ext] v getLogoURIs() !' (S.LogoSet memory)
%E 0x4fe2cf78 [ext] v getLogoIdentity() !' (string memory name_, string memory website_)
%E 0x5125c26f [ext] v getLogoAdmin() !' (address)
%E 0x28bd628f [ext] v getLogoKeeper() !' (address)
%E 0xde9d5062 [ext] v getLogoPorChainIds() !' (uint256[] memory)
%E 0xa3c8b43e [ext] v getLastLogoSyncAt() !' (uint256)
[int] v _readLiquidityState(S.LogoStorage storage st) !' (uint256 coverageBps, bool porFresh)
[int] v _selectState(S.LogoStorage storage st,
uint256 coverageBps,
bool porFresh) !' (uint8 state,
string memory logoURI)

```

Security_back_token/PXON2/CG&CMC/contracts/

USDWMonetarySemanticsFacet.sol

contract USDWMonetarySemanticsFacet (26 fn)

```

%E 0xba4771ba [ext] p assetRole() !' (bytes32)
%E 0x86c9de05 [ext] p assetClass() !' (bytes32)
%E 0xb0c48a4b [ext] p marketParticipation() !' (bool)
%E 0xf4fc7f65 [ext] p upgradeabilityPurpose() !' (bytes32)
%E 0xe8945b66 [ext] p mintModel() !' (bytes32)
%E 0x9acf4c1a [ext] p mintTrigger() !' (bytes32)
%E 0x5c0ae241 [ext] p elasticity() !' (bool)
%E 0x84550aa2 [ext] p tradableOnAMM() !' (bool)
%E 0xdd26b4ed [ext] p sellExpected() !' (bool)
%E 0x9c73ab30 [ext] p transferRole() !' (bytes32)
%E 0x5c46b8e7 [ext] p liquidityModel() !' (bytes32)
%E 0xd77be17a [ext] p liquiditySource() !' (bytes32)
%E 0xfd236ad3 [ext] p dexLiquidityAuthoritative() !' (bool)
%E 0x4a22ad4f [ext] p liquidityType() !' (bytes32)
%E 0x13cf2b06 [ext] p liquidityProof() !' (bytes32)
%E 0x3a0e11c8 [ext] p liquidityAuthoritativeSource() !' (bytes32)
%E 0x33da7e26 [ext] p liquidityFakeable() !' (bool)
%E 0xf7997916 [ext] p marketLiquidityRelevant() !' (bool)
%E 0xf0464826 [ext] p realLiquidityFormula() !' (string memory)
%E 0x35a8ea5d [ext] p emergencyControlModel() !' (bytes32)
%E 0x04caa6dd [ext] p pauseReason() !' (bytes32)
%E 0xc3ee477a [ext] p freezeConditions() !' (bytes32)
%E 0x241e51f8 [ext] p seizure() !' (bool)
%E 0x3f99e657 [ext] p monetarySemanticsBundle() !' (bytes32, bytes32, bytes32, bytes32, bytes32, bytes32, bytes32, bool, bool,
bool)
%E 0x0be245be [ext] p liquidityProtocolBundle() !' (bytes32, bytes32, bytes32, bytes32, bytes32, bytes32, bool, bool, bool)
%E 0x7f4209cb [ext] p monetarySemanticsVersion() !' (string memory)

```

Security_back_token/PXON2/CG&CMC/IUSDWExternals.sol

interface AggregatorV3Interface (5 fn)

```

& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x7284e416 [ext] v description() !' (string memory)
& 0x54fd4d50 [ext] v version() !' (uint256)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId,
int256 answer,
uint256 startedAt
,
uint256 updatedAt,
uint80 answeredInRound)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId,
int256 answer,
uint256 startedAt,
uint256 updatedAt,
uint80 answeredInRound)

```


interface AggregatorV2Interface (5 fn)

```
& 0x50d25bcd [ext] v latestAnswer() !' (uint256)
%É 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%É 0x668a0f02 [ext] v latestRound() !' (uint256)
%É 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (uint256)
%É 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)
```

interface ICoreERC20Facet (4 fn)

```
& 0x18160ddd [ext] v totalSupply() !' (uint256)
& 0x06fddde03 [ext] v name() !' (string memory)
& 0x95d89b41 [ext] v symbol() !' (string memory)
& 0x313ce567 [ext] v decimals() !' (uint8)
```

interface IPriceFeedFacet (2 fn)

```
& 0x93a408d7 [ext] v priceUSD() !' (uint256)
%É 0xffb1a06d [ext] v priceIsStale() !' (bool)
```

interface IPorFacet (1 fn)

```
%É 0x0902f1ac [ext] v getReserves() !' (uint256)
```

interface IPoRMultiChainFacet (1 fn)

```
& 0x688595a0 [ext] v totalReserveUSDE18(uint256[] calldata chainIds) !' (uint256)
```

interface IPorSnapshotFacet (3 fn)

```
& 0x4aafbddb [ext] v isSnapshotFresh(uint256 maxAge) !' (bool)
& 0xd6b8c641 [ext] v getSnapshotTimestamp() !' (uint256)
& 0xa1dae74b [ext] v porSnapshotHash() !' (bytes32)
```

interface IUSDWMetaFacet (1 fn)

```
%É 0x67c859ab [ext] v getPorMaxStaleness() !' (uint256)
```

interface ITokenIdentityFacet (2 fn)

```
& 0x0570df66 [ext] v setTokenLogo(string calldata logoURI)
& 0x291526ac [ext] v tokenLogo() !' (string memory)
```

interface IAutoListFacet (1 fn)

```
%É 0xaa16dda9 [ext] broadcastDiscovery(string calldata name,
    string calldata website,
    string calldata logo)
```

interface IOwnershipFacet (1 fn)

```
& 0x8da5cb5b [ext] v owner() !' (address)
```

Security_back_token/PXON2/CG&CMC/IUSDWMonetarySemantics.sol

interface IUSDWMonetarySemantics (26 fn)

```
%É 0xba4771ba [ext] p assetRole() !' (bytes32)
%É 0x86c9de05 [ext] p assetClass() !' (bytes32)
%É 0xb0c48a4b [ext] p marketParticipation() !' (bool)
%É 0xf4fc7f65 [ext] p upgradeabilityPurpose() !' (bytes32)
%É 0xe8945b66 [ext] p mintModel() !' (bytes32)
%É 0x9acf4c1a [ext] p mintTrigger() !' (bytes32)
%É 0x5c0ae241 [ext] p elasticity() !' (bool)
%É 0x84550aa2 [ext] p tradableOnAMM() !' (bool)
%É 0xdd26b4ed [ext] p sellExpected() !' (bool)
%É 0x9c73ab30 [ext] p transferRole() !' (bytes32)
%É 0x5c46b8e7 [ext] p liquidityModel() !' (bytes32)
%É 0xd77be17a [ext] p liquiditySource() !' (bytes32)
%É 0xfd236ad3 [ext] p dexLiquidityAuthoritative() !' (bool)
%É 0x4a22ad4f [ext] p liquidityType() !' (bytes32)
%É 0x13cf2b06 [ext] p liquidityProof() !' (bytes32)
%É 0x3a0e11c8 [ext] p liquidityAuthoritativeSource() !' (bytes32)
%É 0x33da7e26 [ext] p liquidityFakeable() !' (bool)
%É 0xf7997916 [ext] p marketLiquidityRelevant() !' (bool)
%É 0xf0464826 [ext] p realLiquidityFormula() !' (string memory)
%É 0x35a8ea5d [ext] p emergencyControlModel() !' (bytes32)
%É 0x04caa6dd [ext] p pauseReason() !' (bytes32)
%É 0xc3ee477a [ext] p freezeConditions() !' (bytes32)
%É 0x241e51f8 [ext] p seizure() !' (bool)
%É 0x3f99e657 [ext] p monetarySemanticsBundle() !' (bytes32 assetRole_,
    bytes32 assetClass_,
    bytes32 mintModel_,

    bytes32 transferRole_,
    bytes32 liquidityModel_,
    bytes32 emergencyControlModel_,
    bool tradable0
nAMM_,
    bool dexLiquidityAuthoritative_,
    bool seizure_)
%É 0x0be245be [ext] p liquidityProtocolBundle() !' (bytes32 liquidityType_,
    bytes32 liquidityModel_,
    bytes32 liq
    uiditySource_,
    bytes32 liquidityProof_,
    bytes32 liquidityAuthoritativeSource_,
    bool liquidityFakeable_,

    bool marketLiquidityRelevant_,
    bool dexLiquidityAuthoritative_)
%É 0x7f4209cb [ext] p monetarySemanticsVersion() !' (string memory)
```

Security_back_token/PXON2/CG&CMC/LibUSDWAggregatorStorage.sol

library LibUSDWAggregatorStorage (1 fn)

```
%E 0x86b714e2 [int] p s() !' (LogoStorage storage st)
```

Security_back_token/PXON2/CG&CMC/USDWCGCMCFeedAggregator.sol

contract USDWCGCMCFeedAggregator (23 fn)

```
%E 0x668a0f02 [ext] v latestRound() !' (uint256)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId,
    int256 answer,
    uint256 startedAt
    ,
    uint256 updatedAt,
    uint80 answeredInRound)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId,
    int256 answer,
    uint256 startedAt,
    uint256 updatedAt,
    uint80 answeredInRound)
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%E 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (int256)
%E 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)
%E 0x6406f9db [ext] v getCanonicalSnapshot() !' (CanonicalSnapshot memory snap)
%E 0x22bc2e8a [ext] v updateFeedRound() !' (uint80 roundId, int256 answer)
    [int] v compute() !' (int256 finalPrice8,
    uint256 porPrice8,
    uint256 oraclePrice8,
    uint256 coverageBps,
    uint256 reserveE18,
    uint256 supplyE18,
    uint8 status)
    [int] v _readMultiChainReserve() !' (uint256)
    [int] v _readOraclePrice8() !' (uint256 normalized8)
    [int] _updateVariance(uint256 porPrice8, uint256 oraclePrice8)
%E 0xcc6e478e [ext] setRiskParameters(uint256 maxOracleDeviationBps_,
    uint256 maxPremiumBps_,
    uint256 porStal
    enessOverride_,
    uint256 minOracleVarianceBps_,
    uint256 minPorVarianceBps_,
    uint8 cgIterations_)
%E 0x748747e6 [ext] setKeeper(address keeper_)
%E 0x75829def [ext] transferAdmin(address newAdmin)
%E 0x16c38b3c [ext] setPaused(bool p)
%E 0x2bde8d2a [ext] setMinRefreshInterval(uint256 s_)
%E 0xf97cb290 [ext] setPorChainIds(uint256[] calldata ids)
%E 0x90c3f38f [ext] setDescription(string calldata d)
%E 0x6466afbe [ext] v getPorChainIds() !' (uint256[] memory)
    [prv] p _maxU(uint256 a, uint256 b) !' (uint256)
    [prv] p _sq(uint256 x) !' (uint256)
```

Security_back_token/PXON2/CG&CMC/USDWLogoFromReserveFacet.sol

contract USDWLogoFromReserveFacet (17 fn)

```
%E 0x40753e58 [ext] initializeLogoFacet(uint256[] calldata porChainIds_)
%E 0x2e6057ca [ext] setLogoURIs(string calldata excellent_,
    string calldata healthy_,
    string calldata stable_
    ,
    string calldata warning_,
    string calldata stale_)
%E 0x8424e581 [ext] setLogoIdentity(string calldata name_, string calldata website_)
%E 0xadba0b61 [ext] setLogoKeeper(address keeper_)
%E 0x7977e0a5 [ext] transferLogoAdmin(address newAdmin)
%E 0xa3cffa8b [ext] setLogoPorChainIds(uint256[] calldata ids)
%E 0x896f51cf [ext] setLogoStalenessOverride(uint256 maxAge)
%E 0xfea5272d [ext] syncLogoFromReserve() !' (uint8 newState, string memory newLogo)
%E 0x64f7b224 [ext] v getCurrentLogoState() !' (uint8 state, string memory logoURI, uint256 coverageBps, bool
porFresh)
%E 0x1c1be78c [ext] v getLogoURIs() !' (S.LogoSet memory)
%E 0x4fe2cf78 [ext] v getLogoIdentity() !' (string memory name_, string memory website_)
%E 0x5125c26f [ext] v getLogoAdmin() !' (address)
%E 0x28bd628f [ext] v getLogoKeeper() !' (address)
%E 0xde9d5062 [ext] v getLogoPorChainIds() !' (uint256[] memory)
%E 0xa3c8b43e [ext] v getLastLogoSyncAt() !' (uint256)
    [int] v _readLiquidityState(S.LogoStorage storage st) !' (uint256 coverageBps, bool porFresh)
    [int] v _selectState(S.LogoStorage storage st,
    uint256 coverageBps,
    bool porFresh) !' (uint8 state,
    string memory logoURI)
```

Security_back_token/PXON2/CG&CMC/USDWMonetarySemanticsFacet.sol

contract USDWMonetarySemanticsFacet (26 fn)

```
%E 0xba4771ba [ext] p assetRole() !' (bytes32)
%E 0x86c9de05 [ext] p assetClass() !' (bytes32)
%E 0xb0c48a4b [ext] p marketParticipation() !' (bool)
%E 0xf4fc7f65 [ext] p upgradeabilityPurpose() !' (bytes32)
%E 0xe8945b66 [ext] p mintModel() !' (bytes32)
%E 0x9acf4c1a [ext] p mintTrigger() !' (bytes32)
%E 0x5c0ae241 [ext] p elasticity() !' (bool)
```

```

%Ê 0x84550aa2 [ext] p tradableOnAMM() !' (bool)
%Ê 0xdd26b4ed [ext] p sellExpected() !' (bool)
%Ê 0x9c73ab30 [ext] p transferRole() !' (bytes32)
%Ê 0x5c46b8e7 [ext] p liquidityModel() !' (bytes32)
%Ê 0xd77be17a [ext] p liquiditySource() !' (bytes32)
%Ê 0xfd236ad3 [ext] p dexLiquidityAuthoritative() !' (bool)
%Ê 0x4a22ad4f [ext] p liquidityType() !' (bytes32)
%Ê 0x13cf2b06 [ext] p liquidityProof() !' (bytes32)
%Ê 0x3a0e11c8 [ext] p liquidityAuthoritativeSource() !' (bytes32)
%Ê 0x33da7e26 [ext] p liquidityFakeable() !' (bool)
%Ê 0xf7997916 [ext] p marketLiquidityRelevant() !' (bool)
%Ê 0xf0464826 [ext] p realLiquidityFormula() !' (string memory)
%Ê 0x35a8ea5d [ext] p emergencyControlModel() !' (bytes32)
%Ê 0x04caa6dd [ext] p pauseReason() !' (bytes32)
%Ê 0xc3ee477a [ext] p freezeConditions() !' (bytes32)
%Ê 0x241e51f8 [ext] p seizure() !' (bool)
%Ê 0x3f99e657 [ext] p monetarySemanticsBundle() !' (bytes32, bytes32, bytes32, bytes32, bytes32, bytes32, bool, bool,
bool)
%Ê 0x0be245be [ext] p liquidityProtocolBundle() !' (bytes32, bytes32, bytes32, bytes32, bytes32, bool, bool, bool)
%Ê 0x7f4209cb [ext] p monetarySemanticsVersion() !' (string memory)

```

Security_back_token/PXON2/DexAdapterPluginFacet.sol

contract DexAdapterPluginFacet (7 fn)

```

%Ê 0xf449aac6 [ext] setDexPluginEnabled(bool enabled)
%Ê 0x8a795f8d [ext] setPluginWhitelist(address plugin, bool whitelisted)
%Ê 0xcdd8f01 [ext] setLPWhitelisted(address lpToken, bool whitelisted)
%Ê 0x47154eec [ext] v isDexPluginEnabled() !' (bool)
%Ê 0x52313bc5 [ext] v isPluginWhitelisted(address plugin) !' (bool)
%Ê 0x41e8c4ac [ext] v isLPWhitelisted(address lpToken) !' (bool)
%Ê 0xebf09902 [ext] recordDexAction(bytes32 action, uint256 amount, address token)

```

Security_back_token/PXON2/DexRouterFacet.sol

contract DexRouterFacet (8 fn)

```

%Ê 0xc0d78655 [ext] setRouter(address r)
%Ê 0xb0f479a1 [ext] v getRouter() !' (address)
%Ê 0x8187f516 [ext] setPair(address pair)
%Ê 0xe01af92c [ext] setSwapEnabled(bool enabled)
%Ê 0x173865ad [ext] swapAndLiquify(uint256 tokenAmount)
%Ê 0x9cd441da [ext] addLiquidity(uint256 tokenAmount, uint256 ethAmount)
%Ê 0x53d6fd59 [ext] setWhitelist(address account, bool status)
& 0x3af32abf [ext] v isWhitelisted(address account) !' (bool)

```

Security_back_token/PXON2/DiamondInit.sol

contract PXONDiamondInit (2 fn)

```

%Ê 0xb8540af9 [ext] init(InitParams calldata p)
[int] _grantRole(AppStorage storage s, bytes32 role, address account)

```

contract PXONMigrationInit_Phase1 (1 fn)

```

%Ê 0xf3b4e680 [ext] initPhase1(uint256 , bool isBase)

```

contract PXONMigrationInit_Phase2 (1 fn)

```

%Ê 0xaff909b2 [ext] initPhase2(address , uint256)

```

contract PXONMigrationInit_Phase3 (1 fn)

```

%Ê 0x4c9b816d [ext] initPhase3(bytes32 auditHash, string calldata auditUri,
bytes32 wpHash, string calldata wpUri)

```

Security_back_token/PXON2/Diamond.sol

contract DiamondCutFacet (1 fn)

```

%Ê 0xd2157762 [ext] diamondCut(LibDiamond.FacetCut[] calldata _cuts,
address _init,
bytes calldata _calldat
a)

```

contract DiamondLoupeFacet (5 fn)

```

& 0x7a0ed627 [ext] v facets() !' (Facet[] memory f)
& 0xadfca15e [ext] v facetFunctionSelectors(address _f) !' (bytes4[] memory)
& 0x52ef6b2c [ext] v facetAddresses() !' (address[] memory)
& 0xcdffacc6 [ext] v facetAddress(bytes4 _sel) !' (address)
%Ê 0x01ffc9a7 [ext] v supportsInterface(bytes4 _id) !' (bool)

```

contract OwnershipFacet (3 fn)

```

& 0x8da5cb5b [ext] v owner() !' (address)
& 0xf2fde38b [ext] transferOwnership(address _new)
& 0x715018a6 [ext] renounceOwnership()

```

contract ERC165Facet (4 fn)

```

%Ê 0x01ffc9a7 [ext] v supportsInterface(bytes4 id) !' (bool)
& 0x54e7796f [ext] setSupportsInterface(bytes4 id, bool status)
& 0xee41f9d1 [ext] batchSetSupportsInterface(bytes4[] calldata ids, bool[] calldata statuses)
& 0x6660e282 [ext] registerAllPXONInterfaces()

```

Security_back_token/PXON2/LibDiamond.sol

library LibDiamond (12 fn)

```

%Ê 0x9459e9ce [int] p diamondStorage() !' (DiamondStorage storage ds)
%Ê 0xa34d42b8 [int] setContractOwner(address _newOwner)
%Ê 0xce606ee0 [int] v contractOwner() !' (address)
%Ê 0xd1670914 [int] v enforceIsContractOwner()
%Ê 0xa88efd05 [int] diamondCut(FacetCut[] memory _cuts, address _init, bytes memory _calldata)
%Ê 0x32d0daf0 [int] addFunctions(address _facet, bytes4[] memory _sels)

```

```

%Ê 0x1afb9526 [int] replaceFunctions(address _facet, bytes4[] memory _sels)
%Ê 0xf203661a [int] removeFunctions(address _facet, bytes4[] memory _sels)
%Ê 0xaf679266 [int] addFacet(DiamondStorage storage ds, address _facet)
%Ê 0xaf679266 [int] removeFunction(DiamondStorage storage ds, address _facet, bytes4 _sel)
%Ê 0x5e725d19 [int] initializeDiamondCut(address _init, bytes memory _calldata)
%Ê 0x474d5287 [int] v enforceHasCode(address _c, string memory _msg)

```

Security_back_token/PXON2/LibShared.sol

library LibAccessControl (4 fn)

```

& 0x91d14854 [int] v hasRole(bytes32 role, address account) !' (bool)
%Ê 0xfaf27024 [int] v requireRole(bytes32 role)
& 0x2f2ff15d [int] grantRole(bytes32 role, address account)
& 0xd547741f [int] revokeRole(bytes32 role, address account)

```

library LibTokenOps (3 fn)

```

& 0x40c10f19 [int] mint(address to, uint256 amount)
& 0x9dc29fac [int] burn(address from, uint256 amount)
%Ê 0xab7fece2 [int] transferUnchecked(address from, address to, uint256 amount)

```

library LibCompliance (3 fn)

```

& 0xb9209e33 [int] v isVerified(address investor) !' (bool)
%Ê 0xa02ff685 [int] v canTransfer(address from, address to, uint256 amount, uint256 seriesId) !' (bool, bytes1, bytes32)
%Ê 0x73ac83ef [int] recordTransfer(address from, uint256 amount)

```

library LibPoR (2 fn)

```

%Ê 0xc9a396e9 [int] v getReserve(address feed) !' (uint256 reserveE8, uint256 updatedAt)
& 0x18b9be2e [int] v checkMintPolicy(uint256 mintAmount) !' (bool ok, string memory reason)

```

library LibSafeToken (3 fn)

```

%Ê 0xd1660f99 [int] safeTransfer(address token, address to, uint256 amount)
%Ê 0xd9fc4b61 [int] safeTransferFrom(address token, address from, address to, uint256 amount)
%Ê 0xf788baec [int] v balanceOf(address token, address account) !' (uint256)

```

library LibPeg (2 fn)

```

%Ê 0x2073b815 [int] v getPriceE18(address feed) !' (uint256)
%Ê 0xeb508a60 [int] p isPegHealthy(uint256 priceE18, uint256 minE18, uint256 maxE18) !' (bool)

```

Security_back_token/PXON2/PriceFeedFacetBSC.sol

interface IDiamondView (2 fn)

```

& 0x18160ddd [ext] v totalSupply() !' (uint256)
%Ê 0x0902f1ac [ext] v getReserves() !' (uint256)

```

contract PriceFeedFacetBSC (16 fn)

```

[int] p _s() !' (PFStorage storage s)
& 0x93a408d7 [ext] p priceUSD() !' (uint256)
& 0x9a105d2b [ext] p pegTargetE18() !' (uint256)
& 0xd7355955 [ext] p isPegged() !' (bool)
& 0x313ce567 [ext] p decimals() !' (uint8)
& 0x7284e416 [ext] p description() !' (string memory)
& 0x54fd4d50 [ext] p version() !' (uint256)
& 0x50d25bcd [ext] p latestAnswer() !' (int256)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt, uint256 updatedAt, uint80 answeredInRound)
& 0xfd915e6d [ext] v reserveBacking() !' (uint256)
& 0xb8f01c3b [ext] v liquidityUSD() !' (uint256)
& 0x52d3f41a [ext] v marketCapUSD() !' (uint256)
& 0x9358928b [ext] v circulatingSupply() !' (uint256)
& 0xb86e7822 [ext] v feedSummary() !' (FeedSummary memory f)
& 0x5853b613 [ext] touchFeed()
& 0x6301a747 [ext] p priceFeedVersion() !' (string memory)

```

Security_back_token/PXON2/PriceLogicFacet.sol

library LibPriceStorage (1 fn)

```

%Ê 0x1cc0c06d [int] p priceStorage() !' (PriceState storage ps)

```

interface IPegPolicyFacet (2 fn)

```

& 0x62e0bdbb [ext] v assertPegE18(uint256 priceE18) !' (bool)
& 0x7cf981f2 [ext] v pegMinE18() !' (uint256)

```

contract PriceLogicFacet (7 fn)

```

& 0xc97ab421 [ext] addPriceSource(address source)
& 0x623b6549 [ext] removePriceSource(address source)
& 0x8d6cc5da [ext] updatePrice(uint256 priceE18)
& 0x98d5fdca [ext] v getPrice() !' (uint256)
& 0xbb2ec9af [ext] v getPriceSources() !' (address[] memory)
& 0xb7f88320 [ext] v getLastUpdateTime() !' (uint256)
& 0x78bc2ff9 [ext] v isPriceSource(address source) !' (bool)

```

Security_back_token/PXON2/PriceSourcesFacet.sol

contract PriceSourcesFacet (3 fn)

```

& 0xc97ab421 [ext] addPriceSource(address f)
& 0x623b6549 [ext] removePriceSource(address f)
& 0x66da0e2d [ext] v priceSources() !' (address[] memory)

```

Security_back_token/PXON2/RealDataFacet.sol

interface IChainlinkFeed (3 fn)

```
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
& 0xfef9f968c [ext] v latestRoundData() !' (uint80, int256, uint256, uint256, uint80)
& 0x313ce567 [ext] v decimals() !' (uint8)
```

interface IPoR (3 fn)

```
%E 0x1ef93abb [ext] v getLastProof() !' (bytes32)
& 0xdb249f10 [ext] v getMinReserveRatio() !' (uint256)
%E 0x00873b5 [ext] v isMintingDisabledByPoR() !' (bool)
```

interface IERC20Supply (2 fn)

```
& 0x18160ddd [ext] v totalSupply() !' (uint256)
& 0x70a08231 [ext] v balanceOf(address) !' (uint256)
```

library RealDataLib (1 fn)

```
%E 0x86b714e2 [int] p s() !' (S storage ds)
```

contract RealDataFacet (15 fn)

```
& 0x7406f76b [ext] initRealData(address oracle_,
    address porContract_,
    string calldata logoURI_,
    string calldata coingeckoId_,
    string calldata cmcSlug_)
& 0x12d0b57c [ext] recordSwap(address from,
    address to,
    uint256 amount,
    bool isSwap)
& 0xa126f13a [ext] snapshotHourly()
& 0xb745ae93 [ext] v getRealMetrics() !' (int256 priceE8,
    uint256 supplyUSDW,
    uint256 porUSD,
    uint256 volume1h,
    uint256 volume24h,
    uint256 volume7d,
    uint256 volume30d,
    uint256 volumeAllTime,
    uint256 swapVolume24h,
    uint256 txCount24h,
    uint256 uniqueAddresses24h,
    uint256 updatedAt,
    bytes32 p
    orProofHash,
    bool porHealthy)
& 0x4ad17cdc [ext] v getPriceHistory(uint16 count) !' (uint256[] memory timestamps,
    uint256[] memory prices,
    uint256[] memory supplies,
    uint256[] memory pors,
    uint256[] memory volumes)
& 0x6c02a931 [ext] v tokenName() !' (string memory)
& 0x7b61c320 [ext] v tokenSymbol() !' (string memory)
& 0x8820a08e [ext] v tokenLogoURI() !' (string memory)
& 0x3defeb04 [ext] v tokenWebsite() !' (string memory)
& 0xc2fe0fd8 [ext] v coingeckoId() !' (string memory)
& 0x80e0704b [ext] v cmcSlug() !' (string memory)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
& 0xc30ff6df [ext] setLogoURI(string calldata uri)
& 0x4a2ff7d73 [ext] setCoingeckoId(string calldata id)
& 0xf968dfdd [ext] setCMCSlug(string calldata slug)
```

Security_back_token/PXON2/TokenDataInit.sol

library LibTokenData (1 fn)

```
%E 0x3408f73a [int] p getStorage() !' (TokenStorage storage ds)
```

contract TokenDataInit (3 fn)

```
%E 0xe1c7392a [ext] init()
%E 0x6addb663 [ext] p tokenInfo() !' (string memory name,
    string memory symbol,
    uint8 decimals_,
    uint256 priceUSD6,
    string memory website_,
    string memory logo,
    string memory wp)
%E 0x5c6a29f3 [ext] p reserveInfo() !' (uint256 reserveUSD6,
    string memory account,
    string memory txId)
```

Security_back_token/PXON2/TokenIdentityFacet.sol

library TokenIdentityLib (1 fn)

```
%E 0x86b714e2 [int] p s() !' (S storage ds)
```

contract TokenIdentityFacet (10 fn)

```
& 0xae8a51ab [ext] initTokenIdentity(string calldata logoDataURI_,
    address priceFeed_)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
& 0x291526ac [ext] v tokenLogo() !' (string memory)
& 0x0570df66 [ext] setTokenLogo(string calldata logoDataURI_)
& 0x7b96b8e0 [ext] v getTokenIdentity() !' (Identity memory id)
```



```

    & 0xd7f24abd [ext] v getChainDeployment(uint256 index) !' (uint256 chainId, string memory chainName, address
diamond, uint32 lz
    Eid)
    & 0xd25d23bd [ext] v getChainCount() !' (uint256)
    & 0xeb6db8f5 [ext] addChainDeployment(uint256 chainId, string calldata chainName, address diamond, uint32 lzEid)
    & 0x0a8f41b4 [ext] v getSocialLinks() !' (string memory website, string memory twitter,
        string memory telegram, string
        memory github,
        string memory whitepaper)
    & 0x995d9ab7 [ext] v getTags() !' (string[] memory)

```

Security_back_token/PXON2/UniversalPegGuard.sol

interface IAMM_V2 (3 fn)

```

%E 0x0902f1ac [ext] v getReserves() !' (uint112, uint112, uint32)
%E 0x0dfe1681 [ext] v token0() !' (address)
%E 0xd21220a7 [ext] v token1() !' (address)

```

interface IAMM_V3 (4 fn)

```

%E 0x3850c7bd [ext] v slot0() !' (uint160 sqrtPriceX96, int24 tick,
    uint16, uint16, uint16, uint8, bool)
%E 0x0dfe1681 [ext] v token0() !' (address)
%E 0xd21220a7 [ext] v token1() !' (address)
%E 0x1a686502 [ext] v liquidity() !' (uint128)

```

interface IPriceOracle (1 fn)

```

& 0x50d25bcd [ext] v latestAnswer() !' (int256)

```

library UniversalPegGuardLib (1 fn)

```

%E 0x86b714e2 [int] p s() !' (Storage storage ds)

```

contract UniversalPegGuard (14 fn)

```

& 0x77f62e04 [ext] initUniversalGuard(address usdwToken,
    address oracle,
    uint256 maxBandBps_,
    uint25
    6 minLiquidityUsd)
& 0x703b61cb [ext] setMaxBandBps(uint256 bps)
& 0x3ea6e071 [ext] setExemption(address account, bool exempt)
& 0x2e3cfe49 [ext] setGuardPaused(bool paused)
& 0x8e40b35e [ext] v enforceInvariant(address from,
    address to,
    uint256 amount)
    [int] v _isAMM(address addr,
    UniversalPegGuardLib.Storage storage st) !' (bool)
    [int] v _staticProbe(address addr, bytes4 selector) !' (bool success)
    [int] v _checkPoolPrice(address pool,
    uint256 amount,
    bool incoming,
    UniversalPegGuardL
    ib.Storage storage st)
    [int] v _checkV2Price(address pool,
    uint256 amount,
    bool incoming,
    UniversalPegGuardLib
    .Storage storage st)
    [int] v _checkV3Price(address pool,
    uint256 amount,
    bool incoming,
    UniversalPegGuardLib
    .Storage storage st)
    [int] v _assertPriceInBand(address pool,
    uint256 priceE8,
    uint256 amount,
    UniversalPegGuar
    dLib.Storage storage st)
& 0x48334596 [ext] v isAMMAddress(address addr) !' (bool)
& 0x2cedb376 [ext] v getGuardConfig() !' (uint256 maxBandBps,
    bool paused,
    address oracle)
    [int] p _uintToStr(uint256 v) !' (string memory)

```

Security_back_token/PXON2/UpgradeabilityCompatFacet.sol

contract UpgradeabilityCompatFacet (3 fn)

```

& 0x5c60da1b [ext] v implementation() !' (address)
& 0x3659cfe6 [ext] p upgradeTo(address)
& 0x4f1ef286 [ext] p upgradeToAndCall(address,bytes calldata)

```

Security_back_token/PXON2/USDWIdentityBeacon.sol

contract USDWIdentityBeacon (7 fn)

```

%E 0xeeb72866 [ext] p identify() !' (string memory tokenType,
    string memory symbol,
    uint8
    tokenDecimals,

    string memory backing,
    string memory pegAsset,
    string memory pegRatio,
    bool
    isAlgorithmi
    c,
    bool
    isTrading,
    bool
    isPegged,
    uint256
    priceE18,
    string memory issuer,
    string memory porPortal,
    string memory tokenURI_)

```



```

%Ê 0x974d1e26 [ext] p allDiamonds() !' (address bsc, address polygon, address arbitrum)
%Ê 0x4cf701a3 [ext] p allPriceFeeds() !' (address bsc, address polygon, address arbitrum)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80, int256, uint256, uint256, uint80)
& 0x313ce567 [ext] p decimals() !' (uint8)
& 0x7284e416 [ext] p description() !' (string memory)
%Ê 0x05957706 [ext] v getEngine() !' (address)

```

Security_back_token/PXON2/USDWListingRegistry.sol

contract USDWListingRegistry (12 fn)

```

[prv] p _addChain(ChainRecord memory r)
%Ê 0x42cdbc1 [ext] setChainRecord(uint256 chainId,
    address diamond,
    address proxy,
    address aggregato
r,
    uint16 phaseId,
    bool active)
%Ê 0xb8d217f1 [ext] setSupply(uint256 total, uint256 max, uint256 reserves)
%Ê 0x2c0b69c2 [ext] setLinks(string calldata _website, string calldata _whitepaper,
    string calldata _lo
go, string calldata _por)
%Ê 0x2b596f6d [ext] lockRegistry()
%Ê 0x9217c438 [ext] v getOracleAddress(uint256 chainId) !' (address proxy, uint16 phaseId)
%Ê 0x67ccdf38 [ext] v getTokenAddress(uint256 chainId) !' (address)
%Ê 0x1d776323 [ext] v getChainIds() !' (uint256[] memory)
%Ê 0x9ca220dd [ext] v getAllChains() !' (ChainRecord[] memory records)
%Ê 0xbb51702f [ext] v getListingMetadata() !' (string memory name,
    string memory symbol,
    string memory category,

    string memory peg,
    uint8 decimals_,
    uint256 totalSupply_,
    uint256 maxSupply_,
    uint256 re
serves_,
    string memory website_,
    string memory whitepaper_,
    string memory logoUrl_,
    string memory
porDashboard_)
%Ê 0x68ac3b18 [ext] p getOracleSpec() !' (string memory standard,
    string memory functionSig,
    uint8 decimals_,

    int256 expectedAnswer,
    string memory humanReadable)
%Ê 0x05e66647 [ext] v getCollateralRatioBps() !' (uint256)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/diamond/DiamondCutFacet.sol

contract DiamondCutFacet (1 fn)

```

%Ê 0xd2157762 [ext] diamondCut(LibDiamond.FacetCut[] calldata _cut,
    address _init,
    bytes calldata _calldata)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/diamond/DiamondLoupeFacet.sol

contract DiamondLoupeFacet (4 fn)

```

& 0x7a0ed627 [ext] v facets() !' (Facet[] memory facets_)
& 0xadfca15e [ext] v facetFunctionSelectors(address _facet) !' (bytes4[] memory)
& 0x52ef6b2c [ext] v facetAddresses() !' (address[] memory)
& 0xcdffacc6 [ext] v facetAddress(bytes4 _selector) !' (address)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/GuardianFacet.sol

contract GuardianFacet (13 fn)

```

[prv] p _unfreezeStore() !' (UnfreezeStore storage u)
%Ê 0xa526d83b [ext] addGuardian(address guardian)
%Ê 0x71404156 [ext] removeGuardian(address guardian)
& 0xe60211dc [ext] setQuorum(uint16 newQuorum)
%Ê 0x8debb77 [ext] freezeWallet(string calldata reason)
%Ê 0x926fb3ed [ext] proposeUnfreeze() !' (uint256 id)
%Ê 0x77882110 [ext] approveUnfreeze(uint256 id)
[prv] _maybeFinalize(uint256 id)
%Ê 0x0c68ba21 [ext] v isGuardian(address a) !' (bool)
%Ê 0x0665f04b [ext] v getGuardians() !' (address[] memory)
%Ê 0xc26c12eb [ext] v getQuorum() !' (uint16)
%Ê 0x33eeb147 [ext] v isFrozen() !' (bool frozen,
    address frozenBy,
    uint64 frozenAt)
%Ê 0x1bbf6c8f [ext] v getUnfreezeProposal(uint256 id) !' (uint64 proposedAt,
    uint16 approvalCount,
    bool resolv
ed)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/ KeyRotationFacet.sol

contract KeyRotationFacet (7 fn)

```
%E 0x1c6fd2ca [ext] proposeRotation(address newOwner) !' (uint256 id)
%E 0xb9080d42 [ext] approveRotation(uint256 id)
%E 0x7b7d322d [ext] executeRotation(uint256 id)
%E 0x7cecaf88 [ext] cancelRotation(uint256 id)
%E 0x4e89386e [ext] v getCurrentRotation() !' (uint256 id,
    address newOwner,
    uint64 proposedAt,
    uint16
    approvalCount,
    uint16 required,
    bool executed,
    bool cancelled,
    bool expired)
%E 0x0175316e [ext] v hasApprovedRotation(uint256 id, address who) !' (bool)
%E 0x893d20e8 [ext] v getOwner() !' (address)
```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/MultiSigFacet.sol

contract MultiSigFacet (8 fn)

```
%E 0x2c3f00c9 [ext] setLargeTxThreshold(address token, uint256 threshold)
%E 0x43daa836 [ext] setMultiSigExpiry(uint64 newSeconds)
%E 0x64a8a0a0 [ext] proposeTransfer(address token, address to, uint256 amount) !' (uint256 id)
%E 0xcacca4a0 [ext] approveTransfer(uint256 id)
%E 0xf05d5752 [ext] executeTransfer(uint256 id)
%E 0x59caecb5 [ext] cancelTransfer(uint256 id)
%E 0xf91a7b89 [ext] v getPendingTransfer(uint256 id) !' (address token,
    address to,
    uint256 amount,
    uint
    64 proposedAt,
    uint16 approvalCount,
    uint16 required,
    bool executed,
    bool cancelled,
    bool expired)
%E 0x2358d5a8 [ext] v hasApproved(uint256 id, address who) !' (bool)
```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/ ShieldRegistryFacet.sol

contract ShieldRegistryFacet (6 fn)

```
%E 0xa91ee0dc [ext] setRegistry(address registry)
%E 0x4f83f374 [ext] recordShieldEvent(bytes32 eventType, bytes calldata payload)
%E 0x3224152b [ext] anchorExternalSignal(bytes32 signalId,
    bytes32 kind,
    uint8 severity,
    bytes call
    data payload)
%E 0x5ab1bd53 [ext] v getRegistry() !' (address)
%E 0x9c93ef96 [ext] v getWalletConfig() !' (address owner,
    address registry,
    bool frozen,
    uint64 whi
    telistTimeLock,
    uint64 multiSigExpiry,
    uint16 guardianCount,
    uint16 quorum)
& 0x54fd4d50 [ext] p version() !' (string memory)
```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/ SpendLimitFacet.sol

contract SpendLimitFacet (8 fn)

```
%E 0x2803212f [ext] setDailyLimit(address token, uint256 limit)
%E 0xbeabacc8 [ext] transfer(address token, address to, uint256 amount)
%E 0x7d2e90c2 [ext] transferNative(address payable to, uint256 amount)
%E 0x58f2ef17 [ext] v getDailyLimit(address token) !' (uint256)
%E 0x5b8c9337 [ext] v getDailySpent(address token) !' (uint256 spent,
    uint256 remaining,
    uint64 windowStart)
%E 0xe46638e6 [ext] v canTransfer(address token, address to, uint256 amount) !' (bool)
%E 0x827c4c05 [ext] resetWindow(address token)
[prv] _checkAndAccrue(address token, address to, uint256 amount)
```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/facets/ WhitelistTimelockFacet.sol

contract WhitelistTimelockFacet (9 fn)

```
%E 0x26e192f0 [ext] proposeDestination(address dest) !' (uint64 activeAt)
%E 0x7d6e46a1 [ext] addImmediateDestination(address dest)
%E 0x36d29c18 [ext] cancelPendingDestination(address dest)
%E 0x5cadd08d [ext] removeDestination(address dest)
%E 0xf05a010c [ext] setTimelockSeconds(uint64 newSeconds)
& 0x3af32abf [ext] v isWhitelisted(address dest) !' (bool)
%E 0x917b9f14 [ext] v getDestination(address dest) !' (uint64 enabledAt,
    bool active,
```

```

        bool    consensusAdded,

        uint64 secondsRemaining)
%Ê 0xdac3f78e [ext] v getTimelockSeconds() !' (uint64)
%Ê 0xb9ee259de [ext] v timeUntilActive(address dest) !' (uint64)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/init/ShieldInit.sol

contract ShieldInit (1 fn)

```

%Ê 0x8689ce4a [ext] initGuardedWallet(address owner,
    address[] calldata guardians,
    uint16 quorum,

    address usdwToken,
    uint256 dailyLimit,
    uint256 largeThreshold)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/interfaces/IShieldShared.sol

interface IERC20Min (2 fn)

```

& 0xa9059cbb [ext] transfer(address to, uint256 amount) !' (bool)
& 0x70a08231 [ext] v balanceOf(address who) !' (uint256)

```

interface IShieldRegistry (1 fn)

```

%Ê 0x47f5ad72 [ext] recordEvent(address wallet,
    bytes32 eventType,
    bytes calldata payload)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/libraries/LibDiamond.sol

library LibDiamond (8 fn)

```

%Ê 0x9459e9ce [int] p diamondStorage() !' (DiamondStorage storage ds)
%Ê 0xa34d42b8 [int] setContractOwner(address _newOwner)
%Ê 0xce606ee0 [int] v contractOwner() !' (address)
%Ê 0xd1670914 [int] v enforceIsContractOwner()
%Ê 0xa88efd05 [int] diamondCut(FacetCut[] memory _cut,
    address _init,
    bytes memory _calldata)
%Ê 0x993aa968 [prv] _addFunctions(DiamondStorage storage ds,
    address facet,
    bytes4[] memory selectors)
%Ê 0x560b84db [prv] _replaceFunctions(DiamondStorage storage ds,
    address facet,
    bytes4[] memory selectors)
%Ê 0x25e05d0f [prv] _removeFunctions(DiamondStorage storage ds,
    bytes4[] memory selectors)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/libraries/LibShieldStorage.sol

library LibShieldStorage (1 fn)

```

%Ê 0x86b714e2 [int] p s() !' (Layout storage l)

```

Security_back_token/PXON2/USDW-SHIELD-001/contracts/usdw-directory/ShieldDirectoryFacet.sol

interface IShieldGuardedWallet (1 fn)

```

%Ê 0x893d20e8 [ext] v getOwner() !' (address)

```

contract ShieldDirectoryFacet (7 fn)

```

%Ê 0x36ea1b7b [ext] registerShieldWallet(address shieldwallet)
%Ê 0xb57a7277 [ext] deregisterShieldWallet()
%Ê 0x8ba79e22 [ext] v shieldWalletOf(address user) !' (address)
%Ê 0xc38c7b64 [ext] v shieldOwnerOf(address shieldwallet) !' (address)
%Ê 0x82395b41 [ext] v hasShield(address user) !' (bool)
%Ê 0x06271336 [ext] v totalShieldsRegistered() !' (uint256)
%Ê 0xa333140a [ext] p shieldDirectoryVersion() !' (string memory)

```

Security_back_token/PXON2/USDWShield.sol

interface IAMM_V2 (4 fn)

```

%Ê 0x0902f1ac [ext] v getReserves() !' (uint112, uint112, uint32)
%Ê 0x0dfe1681 [ext] v token0() !' (address)
%Ê 0xd21220a7 [ext] v token1() !' (address)
& 0x18160ddd [ext] v totalSupply() !' (uint256)

```

interface ITWAP (2 fn)

```

%Ê 0x5909c0d5 [ext] v price0CumulativeLast() !' (uint256)
%Ê 0x5a3d5493 [ext] v price1CumulativeLast() !' (uint256)

```

interface IChainlinkFeed (2 fn)

```

& 0x50d25bcd [ext] v latestAnswer() !' (int256)
& 0x313ce567 [ext] v decimals() !' (uint8)

```

library ShieldStorage (1 fn)

```

%Ê 0x86b714e2 [int] p s() !' (S storage ds)

```

contract USDWShield (18 fn)

```
%E 0xadd4d4b1 [ext] initShield(address usdwToken,
    address oracle,
    address[] calldata allowedQuoteTokens_)
%E 0x1cc9f326 [ext] shield(address pool, uint256 amount, address caller)
    [int] _flashLoanGuard(ShieldStorage.S storage st, address caller)
    [int] _sandwichGuard(ShieldStorage.S storage st, address pool)
    [int] _twapGuard(ShieldStorage.S storage st, address pool)
    [int] _liquidityGuard(ShieldStorage.S storage st, address pool)
    [int] _swapImpactGuard(ShieldStorage.S storage st,
        address pool,
        uint256 amount)
    [int] _donationGuard(ShieldStorage.S storage st, address pool)
    [int] v _getPoolReserves(address pool, address usdwToken) !' (uint256 resUSDW, uint256 resQuote)
    [int] _indirectPairGuard(ShieldStorage.S storage st, address pool)
%E 0x487a2ad6 [ext] queueConfigChange(bytes32 actionHash)
%E 0xc996dd86 [ext] executeConfigChange(string calldata param,
    uint256 newValue,
    uint256 queuedAt)
    [int] _applyConfigChange(ShieldStorage.S storage st,
        string calldata param,
        uint256 value)
    [int] v _getPoolSpotE18(address pool, address usdwToken) !' (uint256 priceE18)
%E 0xa90db408 [ext] setAllowedQuoteToken(address token, bool allowed)
%E 0x4b0bddd2 [ext] setAdmin(address account, bool granted)
%E 0xf5d53a36 [ext] setGuardsActive(bool active)
%E 0xc3f909d4 [ext] v getConfig() !' (uint256 maxBandBps,
    uint256 maxBlockPriceMoveBps,
    uint256 twapWindowSec,
    uint256 maxTwapDevBps,
    uint256 minLiquidityUsdWei,
    uint256 maxSwapImpactBps,
    uint256 timelockDelayS
    ec,
    bool allGuardsActive)
```

Security_back_token/PXON2/USDWUSDAggregator.sol

interface AggregatorInterface (5 fn)

```
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%E 0x668a0f02 [ext] v latestRound() !' (uint256)
%E 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (int256)
%E 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)
```

interface AggregatorV3Interface (5 fn)

```
& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x7284e416 [ext] v description() !' (string memory)
& 0x54fd4d50 [ext] v version() !' (uint256)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId, int256 answer, uint256 startedAt, uint256
updatedAt, uint80 answeredInRound)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt, uint256 updatedAt,
uint80 answeredInRound)
```

abstract contract Owned (2 fn)

```
& 0xf2fde38b [ext] transferOwnership(address _to)
%E 0x79ba5097 [ext] acceptOwnership()
```

contract USDWUSDAggregator (9 fn)

```
%E 0xa87a20ce [ext] updateAnswer(int256 _answer)
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%E 0x668a0f02 [ext] v latestRound() !' (uint256)
%E 0xb5ab58dc [ext] v getAnswer(uint256 _roundId) !' (int256)
%E 0xb633620c [ext] v getTimestamp(uint256 _roundId) !' (uint256)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80 roundId, int256 answer, uint256 startedAt, uint256
updatedAt, uint80 answeredInRound)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt, uint256 updatedAt,
uint80 answeredInRound)
    [int] _recordRound(int256 _answer)
```

Security_back_token/PXON2/USDWWalletMetadata.sol

contract USDWWalletMetadata (12 fn)

```
& 0x06fddde03 [ext] p name() !' (string memory)
& 0x95d89b41 [ext] p symbol() !' (string memory)
& 0x313ce567 [ext] p decimals() !' (uint8)
%E 0x3c130d90 [ext] p tokenURI() !' (string memory)
& 0xe8a3d485 [ext] p contractURI() !' (string memory)
%E 0x6bb38b28 [ext] p logoURI() !' (string memory)
%E 0x7c233b04 [ext] p logoSVG() !' (string memory)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt,
uint256 updatedAt, uint80
answeredInRound)
%E 0x0ce0ca1c [ext] p decimalsPrice() !' (uint8)
& 0x7284e416 [ext] p description() !' (string memory)
%E 0xa1d52c5d [ext] p walletInfo() !' (WalletDisplayInfo memory info)
%E 0x01ffc9a7 [ext] p supportsInterface(bytes4 id) !' (bool)
```

Security_back_token/PXON2/USDW_WALLET_VISIBILITY/ USDWMetadataBridgeFacet.sol

interface IUSDWOnChainResolver (2 fn)

```
%E 0x3c130d90 [ext] v tokenURI() !' (string memory)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
```

library LibMetadataBridgeStorage (1 fn)

```
%E 0x86b714e2 [int] p s() !' (BridgeStorage storage st)
```

contract USDWMetadataBridgeFacet (7 fn)

```
%E 0x7d9e9ffe [ext] initializeMetadataBridge(address resolver_)
%E 0x3c130d90 [ext] v tokenURI() !' (string memory)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
%E 0x7995c1e4 [ext] setMetadataResolver(address resolver_)
%E 0xdf1e954d [ext] transferBridgeAdmin(address newAdmin)
%E 0x3c8b6a6f [ext] v getMetadataResolver() !' (address)
%E 0x1970f1ac [ext] v getBridgeAdmin() !' (address)
```

Security_back_token/PXON2/USDW_WALLET_VISIBILITY/ USDWOnChainResolver.sol

interface IUSDWCGCMCFeedAggregator (5 fn)

```
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt,
uint256 updatedAt, uint80
answeredInRound)
& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x7284e416 [ext] v description() !' (string memory)
& 0x54fd4d50 [ext] v version() !' (uint256)
%E 0x6406f9db [ext] v getCanonicalSnapshot() !' (CanonicalSnapshot memory)
```

interface IUSDWLogoFacet (2 fn)

```
%E 0x64f7b224 [ext] v getCurrentLogoState() !' (uint8 state, string memory logoURI, uint256 coverageBps, bool
porFresh)
%E 0x4fe2cf78 [ext] v getLogoIdentity() !' (string memory, string memory)
```

interface IERC165 (1 fn)

```
%E 0x01ffc9a7 [ext] v supportsInterface(bytes4 interfaceId) !' (bool)
```

interface IERC1046 (1 fn)

```
%E 0x3c130d90 [ext] v tokenURI() !' (string memory)
```

interface IERC7572 (1 fn)

```
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
```

contract USDWOnChainResolver (44 fn)

```
%E 0x01ffc9a7 [ext] p supportsInterface(bytes4 interfaceId) !' (bool)
%E 0x3c130d90 [ext] v tokenURI() !' (string memory)
& 0xe8a3d485 [ext] v contractURI() !' (string memory)
%E 0xdb00e6b5 [ext] v getMetadataJSON() !' (string memory)
& 0x313ce567 [ext] v decimals() !' (uint8)
& 0x7284e416 [ext] v description() !' (string memory)
& 0x54fd4d50 [ext] v version() !' (uint256)
& 0xfeaf968c [ext] v latestRoundData() !' (uint80 roundId, int256 answer, uint256 startedAt,
uint256 updatedAt, uint80
answeredInRound)
%E 0x9a6fc8f5 [ext] v getRoundData(uint80 _roundId) !' (uint80, int256, uint256, uint256, uint80)
& 0x50d25bcd [ext] v latestAnswer() !' (int256)
%E 0x8205bf6a [ext] v latestTimestamp() !' (uint256)
%E 0x668a0f02 [ext] v latestRound() !' (uint256)
%E 0xb5ab58dc [ext] v getAnswer(uint256 roundId) !' (int256)
%E 0xb633620c [ext] v getTimestamp(uint256 roundId) !' (uint256)
%E 0xb4c96aeb [ext] v getFullMetadata() !' (FullMetadata memory m)
%E 0x330227bf [ext] v getPriceInfo() !' (int256 priceE8, uint8 priceDecimals_, uint256 updatedAt,
uint8 statusCode, stri
ng memory statusLabel)
%E 0xffe777bd [ext] v getReserveInfo() !' (uint256 reserveE18, uint256 supplyE18, uint256 coverageBps,
bool porFresh, ui
nt256[] memory chainIds)
%E 0xf000b97b [ext] v getLogoInfo() !' (uint8 state, string memory logoURI, uint256 coverageBps, bool porFresh)
& 0x06fddde03 [ext] v name() !' (string memory)
& 0x95d89b41 [ext] v symbol() !' (string memory)
& 0x18160ddd [ext] v totalSupply() !' (uint256)
%E 0x2e983054 [ext] setProjectInfo(ProjectInfoInput calldata p)
%E 0x36eb5364 [ext] setAuditInfo(string calldata url_,
string calldata standard_,
string calldata auditor_)
'
uint256 dateUnix_,
bytes32 reportHash_)
%E 0x18023950 [ext] setIssuerInfo(string calldata name_,
string calldata legalId_,
string calldata country_)
%E 0xcb6442b0 [ext] setCanonicalChainIds(uint256[] calldata ids)
%E 0x75829def [ext] transferAdmin(address newAdmin)
[int] p _statusLabel(uint8 code) !' (string memory)
[int] p _dataURI(string memory json) !' (string memory)
[int] v _metadataJSON() !' (string memory)
[int] v _jsonHead(uint8 tokDec, string memory logoURI_) !' (string memory)
[int] v _jsonPrice(IUSDWCGCMCFeedAggregator.CanonicalSnapshot memory snap, bool snapOk) !' (string
memory)
[int] v _jsonTail(IUSDWCGCMCFeedAggregator.CanonicalSnapshot memory snap,
bool snapOk,
uint8 logoS
tate_) !' (string memory)
```

```

    [int] v _jsonReserves(IUSDWCGCMCFeedAggregator.CanonicalSnapshot memory snap,
bool snapOk) !' (string memor
y)
    [int] v _jsonAudit() !' (string memory)
    [int] v _jsonIssuer() !' (string memory)
    [int] v _jsonWiring(uint8 logoState_) !' (string memory)
    [int] v _chainsJSON() !' (string memory out)
    [int] p _u(uint256 v) !' (string memory)
    [int] p _i(int256 v) !' (string memory)
    [int] p _bool(bool b) !' (string memory)
    [int] p _addrStr(address a) !' (string memory)
    [int] p _b32(bytes32 b) !' (string memory)
    [int] p _hex(uint8 nibble) !' (bytes1)
    [int] p _esc(string memory s) !' (string memory)

```

library Base64 (1 fn)

```

%Ê 0x12496a1b [int] p encode(bytes memory data) !' (string memory)

```


8. Operational Scripts (this report was generated by)

Workspace:	D:\PropertyX\pxon-ops\
enumerate-diamond.js:	READ-ONLY. Rebuilds diamond-state.json from live chain + local Solidity sources.
enumerate-source.js:	READ-ONLY. Rebuilds source-inventory.json by parsing every .sol under Security_back_token/PXON2/.
mint.js:	SIGNS. Sends mint(recipient, amount) via MintBurnFacetProd. Requires DEPLOYER_KEY in .env.
create-series.js:	SIGNS. Iterates plots.json and calls SeriesEngineFacet.createSeries per lot.
generate-report.js:	READ-ONLY. Rebuilds this PDF from JSON artefacts.

None of these scripts hold or transmit your private key outside .env. The user runs signing scripts on their own machine.

9. Security & Compliance Notes

- 1) The mint executes `MintBurnFacetProd.mint`, gated by the `MINTER` role. The signer must currently hold that role; if not, grant it via `RolesFacet.grantRole(MINTER, signer)` first.
- 2) 100,000,000 PXON is the `maxTotalSupply`. After this mint no further PXON can be issued via `mint/issue` paths without first burning; ensure this concentration matches your intended primary-distribution plan.
- 3) `porFeed` is currently `address(0)`. Because `mintDisabledOnPoRFail` was set `false` at init, mint will not be blocked, but the token's PoR panels will show 0 reserve until a Chainlink-compatible feed is registered via `PorFacet.setPorFeed`.
- 4) `plots.json` contains `PLACEHOLDER` cadastral values (deterministic hashes derived from `plotNumber` strings). Real `assetHash/docRoot` values **MUST** be substituted before `create-series.js` is run with `CONFIRM=YES` on mainnet — otherwise the on-chain series metadata will not correspond to any real land title.
- 5) Diamond ownership is currently a single EOA. For production custody, transfer ownership to a multi-sig via `OwnershipFacet.transferOwnership` before wide distribution.