DLT Security Governance Framework

Function	Category	Sub-Category	NIST SP 800-53, Revision 5 Control
PROTECT(PR)	**DLT Smart Contract and Application Security PR.SCAS** **Security policies, processes, and procedures that address DLT application and smart contract security are used to manage the protection of the DLT network's software and related smart contracts. Smart contracts represent real-world legal agreements between the consortia members and enforces the business logic that all DLT transactions must comply with.**	**PR.SCAS-1:** Policies to ensure process to upgrade live smart contracts is robust in preventing potential compromise of business logic execution by malicious actors. e.g., making use of explicit or manual smart contract upgrades vs Implicit or automated upgrades are enforced across all DLT Consortia Member Orgs **PR.SCAS-2:** Policies to validate DApps prevent compromise of business workflow integrity prior to formally approving installation of the DApp on the nodes are enforced across all DLT Consortia Member Orgs (e.g., ensuring DApps are explicitly coded to validate both format and content of all data received from a counterparty against relevant contract prior to processing the transaction) **PR.SCAS-3:** Policies to validate that the business logic flow has been designed as per DLT vendor's "security best practices" with appropriate tools that meet scalability, confidentiality, and privacy requirements of the DLT Consortia's business requirements, prior to formally approving installation of the DApp on the nodes are enforced across all DLT Consortia Member Orgs **PR.SCAS-4:** Policies to prevent inadvertent exposure of confidential business logic workflows to unauthorized counterparties are enforced across all DLT Consortia Member Orgs e.g., keeping the DApp executable and the business workflows separate **PR.SCAS-5:** Policies to validate that the DApp has been analyzed for leaks of PII or proprietary business logic prior to approving installation of the DApp on the nodes are enforced across all DLT Consortia Member Orgs **PR.SCAS-6:** Smart Contract Code Vulnerability Management Process is established, assessed, managed and communicated across all DLT Consortia Member Orgs **PR.SCAS-7:** Audit the automated build process for detecting unauthorized dependencies within the DApp. This control ensures there is an automated review process to detect if an unauthorized dependency is being used. **PR.SCAS-8:** Audit the automated review process of alerts that are sent to an administrative account when the calls to DApps are repetitive or exceed a baseline threshold. This control ensures abnormal client traffic is reviewed. **PR.SCAS-9:** Mechanisms are in place to stop exception and send alerts when unintended flow control or execution is encountered. This control ensures safe exception handling is executed in smart contracts. **PR.SCAS-10:** Mechanisms are in place to explicitly load smart contracts identified and authorized by the transaction across the DLT Consortia Member Org nodes that are participants to the transaction **PR.SCAS-11:** Mechanisms are in place to ensure 1:1 matching between loaded smart contract and transaction contents and constraints **PR.SCAS-12:** Mechanisms are in place to detect and prevent smart contract verification bypass (e.g., validating that the DApps is coded to explicitly throw an exception if an invalid transaction is encountered) **PR.SCAS-13:** Installations, upgrades, and security posture changes should be monitored and logged for forensic investigation. **PR.SCAS-14:** Smart Contract Software is deployed in a secure and decentralized manner to enable Resiliency, Transparency and Censorship Resistance **PR.SCAS-15:** Policies to validate DApps code is free from relevant vulnerabilities that have been identified and published prior to approving installation of the DApp on the nodes are enforced across all DLT Consortia Member Orgs e.g., requiring the DApp to go through extensive vulnerability testing using Static and dynamic application security testing (SAST/DAST) tools **PR.SCAS-16:** Smart Contract Software is monitored for Confidentiality,Integrity, Privacy and Availability compromise **PR.SCAS-17:** Baseline configuration for on-chain section of the smart contract software is established and communicated across all DLT Consortia Member Orgs **PR.SCAS-18: Mandatory standards for securing the decentralized off-chain versions of the smart contract software are established and communicated across all DLT Consortia Member Orgs** **PR.SCAS-19: Smart contracts must be trusted and explicitly installed by the administrator. This control ensures only trusted and tested smart contracts get deployed.** **PR.SCAS-20: Developers should follow engineering security best practices to minimize vulnerabilities during the design and development phase of the development lifecycle. This control ensures Dapps are built with an eye to secure by design and default.** **PR.SCAS-21: Determine baseline thresholds for API calls. Monitor Dapps or APIs for repeated calls that exceed baseline thresholds. This control ensures that the Dapps or API stays unaffected by DoS attacks.** **PR.SCAS-22: Documented Procedures are in place for secure handling of API Keys. Mechanisms are in place to verify that API keys and other credentials are not stored in public-facing source control systems (e.g., GitLab/GitHub). or into firmware, mobile applications, or any client-based application.**	

DLT Security Governance Framework

Function	Category	Sub-Category	NIST SP 800-53, Revision 5 Control
PROTECT (PR)	**DLT User, Node, Network - Identity, Authentication, and Access Control (PR.DUNN-AC)** Access to DLT related assets both logical and physical, ON and OFF chain and associated facilities is limited to authorized credentials of supported identities viz., User, Node and Network and is managed consistent with the assessed risk of unauthorized access to authorized activities and transactions.	**PR.DUNN-AC-1**: Access Credentials for User, Node and Network are bound to valid certificates and secret/private keys and asserted in all interactions including during creation of , signing and encrypting transactions **PR.DUNN-AC-2**: Applicable Trust Requirements for"Indirect Participants" sharing Identity Services with one of the Node hosting organization as a gateway to the DLT Network are identified, established, assessed and managed **PR.DUNN-AC-3**: Secure generation of secret/private keys for User, Node and Network is established, assessed, managed and communicated across all DLT Consortia Member Orgs **PR.DUNN-AC-4**: Secure methods for storing Node and Network's secret/private keys are established, assessed, managed and communicated across all DLT Consortia Member Orgs **e.g.,** Hardware security module (HSM), Trusted Execution Environment (TEE) **PR.DUNN-AC-5**: Secure methods for storing User's secret/private keys are established, assessed, managed and communicated across all DLT Consortia Member Orgs e.g.,non-custodial Hierarchical Deterministic (HD) Wallets with hardened key function (e.g.,Trezor wallets) based on industry standards like BIP-32 or one superseeding it) **PR.DUNN.AC-6**: Access permissions and authorizations are managed, incorporating the principles of least privilege and separation of duties **PR.DUNN-AC-7**: Mechanism is in place to periodically review and revoke expired access credentials for User, Node and Network across all DLT Consortia Member Orgs **PR.DUNN.AC-8**: Remote access into the DLT Network is securely managed across all DLT Consortia Member Orgs **PR.DUNN.AC-9**: DLT Network integrity is protected across all DLT Consortia Member Orgs (e.g., network segregation, network segmentation) **PR.DUNN.AC-10**: Mechanisms are in place to validate the Cipher Suites in use within the DLT Network match the current NIST recommended Quantum Safe Cipher Suites **PR.DUNN.AC-11**: Mechanisms are in place to validate the Cipher Key Lengths in use within the DLT Network comply with NIST recommended minimum Key Length **PR.DUNN.AC-12**: Mechanisms are in place to validate that a rotation algorithm for Cipher Keys has been designated **PR.DUNN.AC-13**: Mechanisms are in place to validate DLT Consortia's approval for Certificate Signing Request (CSR) and Certificate Revocation List (CRL)	/--- --- --- IA-1, IA-2, IA-3, IA-4, IA-5, IA-7, IA-8, IA-9, IA-10, IA-11, IA-12 PE-1, PE-2, PE-3, PE-4, PE-5, PE-6, PE-8, PE-9 AC-1, AC-17, AC-19, AC-20, SC-15 AC-1, AC-2, AC-3, AC-5, AC-6, AC-14, AC-16, AC-24 AC-4, AC-10, SC-7, SC-10, SC-20 AC-16, IA-1, IA-2, IA-4, IA-5, IA-8, IA-12, PE-2, PS-3 AC-14, IA-1, IA-2, IA-3, IA-5, IA-8, IA-9, IA-10, IA-11
DETECT(DE)	**DLT Network/Node Threat Management (DE.DNTM)**	**DE.DNTM-1**: Mechanisms are in place to minimize the high business impact threats to the DLT Network stemming from decentralized DLT network and system administration across DLT Consortia Member Orgs (e.g., engaging a single service provider - preferably a neutral party) **DE.DNTM-2**: Mechanisms are in place to minimize the high business impact threats to the DLT Network stemming from missing Governance Policies for securing DLT network and system administration channels & credentials **DE.DNTM-3**: Mechanisms are in place to minimize the high business impact threats to the DLT Network stemming from compromise of business logic execution and resiliency due to improper business logic flow and DApp design **DE.DNTM-4**: Mechanisms are in place to minimize the high business impact threats to the DLT Network stemming from compromise of Peer Nodes and Vaults **DE.DNTM-5**: Mechanisms are in place to minimize the high business impact threats to the DLT Network stemming from compromise of DLT network and system administration accounts e.g., sending alerts and logging abnormal node, network and DApp admin account activity to a separate "shared admin account" with restricted access **DE.DNTM-6**: Policy of "Seperation of Duties" or "Layered Admin Privilege Role" is enforced at all times to restrict DLT Network, Node, DApps or BNO, Administrator's CLI access to the task on hand **DE.DNTM-7**: Policy of DLT Network, Node or DApps Administrators using authorized mechanisms to access DLT Network/Node is enforced at all times. e.g., Bastion Hosts, Out of Band/Dedicated Channels, Network Isolation, etc. **DE.DNTM-8**: Mechanisms are in place to secure all administrator credentials with elevated privileges at all times (e.g., DLT Network, Node, DApp administrator, Business Network Operator (BNO) credentials in transit, in use and at rest) **e.g.,** 1) with end to end TLS between Client and Nodes as well as between Nodes for in-transit, 2) with enforcing authorized mechanisms for access into the DLT Network like out of band/dedicated channels, bastion hosts etc. for data in use and 3) making use of host native or other secure encryption methods for data at rest) **APIs** **DE.DNTM-9:** Mechanism to minimize the high business impact threats to the DLT Network stemming from decentralized DLT network and system administration across DLT Consortia Member Orgs whenever API key management changes **from** read, to write, to execution of a transaction. **DE.DNTM-10:** Mechanism to minimize the high business impact threats to the DLT Network stemming from the decentralized DLT network and system administration across DLT Consortia Member Orgs whenever an API on any node enables a smart contract to connect to a traditional business application in the Enterprise which is part of the consortia **DE.DNTM-11:** Mechanism to minimize the high business impact threats to the DLT Network stemming from decentralized DLT network and system administration across DLT Consortia Member Orgs whenever an API allows the DLT to be a decentralized authority for external systems that need transactional data & histories.	

DLT Security Governance Framework

Function	Category	Sub-Category	NIST SP 800-53, Revision 5 Control
DETECT(DE)	**Anomalies and Events (DE.DAE)** Anomalous activity within the DLT Network is detected and the potential impact of events is understood	**Peer/Node Alerts:** **DE.DAE-1**: Mechanism to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when untrusted/unvalidated smart contracts are installed on peer/node **DE.DAE-2**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when peer/node configurations deviate from vendor **or ISO** recommended security best practices **DE.DAE-3**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when peer/node exhibits anamalous behavior **e.g.,** Excessive full DB scan; Excessive ledger fetch commands; Large volume of data export; Unexpected changes in configuration files; Abnormal peer metrics values, Outgoing calls made to unauthorized external IP addresses **DE.DAE-4**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when "Seperation of Duties Principle" is violated with regard to node admin host logs i.e. the logs are being accessed by node admin accounts **DE.DAE-5**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when there are unauthorized and unscheduled system related change activities on the peer/node **DE.DAE-6**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when "data-at-rest" scans run on node vaults detect indicators of compromise **DE.DAE-7**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when inbound traffic from the internet to nodes is flagged by firewalls as malicious i.e., not from authorized services and IP addresses e.g., externally hosted Identity or DNS Services **DE.DAE-8**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when outbound traffic from the nodes is not to authorized IP addresses or services e.g., externally hosted Identity or DNS Services **DE.DAE-9**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when unauthorized activity is found in blockchain network configuration folders; **DE.DAE-10**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when unauthorized activity found on peer or consensus nodes **Consensus/Notary Alerts:** **DE.DAE-11**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when consensus/notary node cluster's baseline performance threshold is exceeded **DE.DAE-12**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when when notarisation request sent by node exceeds number of retries **Client Credentials Compromise Alert** **DE.DAE-13**: Mechansim to detect, send alerts and perform a timely review of alerts received is in place at all times across all DLT Consortia Member Orgs when credential databases used for RPC Client logins are compromised. **DE.DAE-14**-Mechanism to detect,send alerts and perform a timely review of all alerts recieved in place across all Consortia member organizations when API key management changes from read to write & execution of a transaction. **DE.DAE-15**-Mechanism to detect,send alerts and perform a timely review of all alerts recieved in place across all Consortia member organizations whenever an API enables a smart contract to connect to a traditional business application in the Consortia Member Orgs Enterprise environment **DE.DAE-16**-Mechanism to detect,send alerts and perform a timely review of all alerts recieved in place across all Consortia member organizations whenever an API allows DLT Network to be decentralized authority for external systems that need transactional data & histories.	