e. Table of Contents

return to Reference Architecture (RA)

- 1 Introduction
  - 1.1 Problem
  - 1.2 Purpose
  - 1.3 Content Organization

2. 2 Architectural Views
   - 2.1 Stakeholder Views
     - 2.1.1 Platform View
     - 2.1.2 Domain View
     - 2.1.3 Ecosystem View
     - 2.1.4 Ecosphere View
     - 2.1.5 Exchange View
     - 2.1.6 Enterprise View
     - 2.1.7 Relevant Community Standards

   2. 2.2 Technical Views
      - 2.2.1 Fundamental Views
        - 2.2.1.1 Interfaces
          - 2.2.1.1.1 Platform Interface
          - 2.2.1.1.2 Software Interfaces
          - 2.2.1.1.3 Human Interfaces

        2. 2.2.1.2 Tools
           - 2.2.1.2.1 Logging
           - 2.2.1.2.2 Semantic Web
           - 2.2.1.2.3 Open Source Communities

        3. 2.2.1.3 Case Management
        4. 2.2.1.4 System of Systems (SoS)
        5. 2.2.1.5 Quality
        6. 2.2.1.6 Open Source Paradigm
        7. 2.2.1.7 Assurance

        2. 2.2.2 Node Network View
           - 2.2.2.1 Network View
             - 2.2.2.1.1 Secure Messaging
             - 2.2.2.1.2 Transport
             - 2.2.2.1.3 Security
             - 2.2.2.1.4 Protocol
             - 2.2.2.1.5 Distribution Software

           2. 2.2.2.2 Node View
              - 2.2.2.2.1 Operating System (OS)
              - 2.2.2.2.2 Operating Environment
              - 2.2.2.2.3 DIDO Platform

        - 2.2.2.2.4 Distributed Applications
    3. 2.2.2.3 Node Architecture
        - 2.2.2.3.1 Immutable Data Objects
            - 2.2.2.3.1.1 Ledger
            - 2.2.2.3.1.2 Transactions
            - 2.2.2.3.1.3 Identities
            - 2.2.2.3.1.4 Wallets
        2. 2.2.2.3.2 Ancillary Data
            - 2.2.2.3.2.1 Journal
            - 2.2.2.3.2.2 Transforms
            - 2.2.2.3.2.3 Distributed Applications
            - 2.2.2.3.2.4 Web Applications
            - 2.2.2.3.2.5 Exchanges
        3. 2.2.2.3.3 Semantic Web
        4. 2.2.2.3.4 Software
    4. 2.2.2.4 Messaging View
    3. 2.2.3 Decentralized Finance (DeFi) Layers
    3. 2.3 Taxonomic Views
        - 2.3.1 Network Topology Taxonomy
            - 2.3.1.1 Centralized Network Topology
            - 2.3.1.2 Decentralized Network Topology
            - 2.3.1.3 Distributed Network Topology
        2. 2.3.2 Network Access Control Taxonomy
            - 2.3.2.1 Permissionless Networks
            - 2.3.2.2 Permissioned Networks
            - 2.3.2.3 Public Networks
            - 2.3.2.4 Private Networks
            - 2.3.2.5 Hybrid Networks
        3. 2.3.3 Node Taxonomy
            - 2.3.3.1 Full Node
                - 2.3.3.1.1 Pruned Node
                - 2.3.3.1.2 Archival Node
                    - 2.3.3.1.2.1 Authority Node
                    - 2.3.3.1.2.2 Staking Node
                    - 2.3.3.1.2.3 Mining Node
                    - 2.3.3.1.2.4 Masternode
            2. 2.3.3.2 Lightweight Node (Wallet)
            3. 2.3.3.3 Lightning Node
            4. 2.3.3.4 Permanode
            5. 2.3.4 Data Taxonomy
                - 2.3.4.3.1 Ledger Data
                - 2.3.4.3.2 Ancillary Data
                - 2.3.4.3.3 External Data
            3. 3 Governance
                - 3.1 DIDO Communities
                    - 3.1.1 Stakeholder Communities
                    - 3.1.2 Software Communities
                2. 3.2 Legal Documents

        - 3.2.1 Charter
        - 3.2.2 Bylaws
        - 3.2.3 Policies and Procedures (P&P)
    3. 3.3 Guides
    4. 4 Requirements
        - 4.1 About Requirements
            - 4.1.1 Governance Requirements Model
            - 4.1.2 Cognitive Requirements Model
            - 4.1.3 Governing Roles - Combined Requirements Model
            - 4.1.4 Example of a Using the Combined Requirements Model
            - 4.1.5 The Current State of DIDO Requirements
            - 4.1.6 One Degree of Freedom Rule
            - 4.1.7 Specifying Requirements
        2. 4.2 Functional Requirements
            - 4.2.1 Platforms
                - 4.2.1.1 Hardware Platform
                - 4.2.1.2 Operating System Platform
                - 4.2.1.3 Runtime Platforms
                - 4.2.1.4 Network Platforms
                - 4.2.1.5 Virtualized Nodes
            2. 4.2.2 Access Control
        3. 4.3 Non-Functional Requirements
            - 4.3.1 Portability
                - 4.3.1.1 Adaptability
                - 4.3.1.2 Installability
                - 4.3.1.3 Replaceability
            2. 4.3.2 Reliability
                - 4.3.2.1 Maturity
                - 4.3.2.2 Availability
                - 4.3.2.3 Fault Tolerance
                - 4.3.2.4 Recoverability
            3. 4.3.3 Maintainability
                - 4.3.3.1 Modularity
                - 4.3.3.2 Reusability
                - 4.3.3.3 Analysability
                - 4.3.3.4 Modifiability
                - 4.3.3.5 Testability
            4. 4.3.4 Securability

- 4.3.4.1 Confidentiality
- 4.3.4.2 Data Integrity
- 4.3.4.3 Non-Repudiation
- 4.3.4.4 Authenticity
- 4.3.4.5 Accountability

5. 4.3.5 Manageability
   - 4.3.5.1 Types of Manageability Functions
   - 4.3.5.2 Manageability Costs
   - 4.3.5.3 System Manageability Issues
   - 4.3.5.4 Software Manageability Issues
6. 4.3.6 Usability
   - 4.3.6.1 Effectiveness Metrics
   - 4.3.6.2 Efficiency Metrics
   - 4.3.6.3 Attitude / Satisfaction Metrics
7. 4.3.7 Performance
   - 4.3.7.1 Platform Performance
   - 4.3.7.2 Application Performance
   - 4.3.7.3 Network Performance
8. 4.3.8 Interoperability
9. 4.3.9 Elasticity
10. 4.3.10 Scalability

4. 4.4.2.1 How to Use the Non-Functional Requirements Boilerplate
5. 4.4 Assessing Requirements
   - 4.4.1 Functional Requirements Assessment
   - 4.4.2 Non-functional Requirements Assessment

- Appendix A: Glossary of Terms Related to DIDO
- Appendix B: Standards Organizations
- Appendix C: Hardware Architectures
- Appendix D: Operating Systems
- Appendix E: Tools
- Appendix F: DDS Quality Of Service
- Appendix G: Tests
- Appendix H: Acronyms
- Appendix I: Cognitive Model
- Appendix J: Governance Model

- Parking Lot

☐ [char]Update with new sections

