SSI Deployment Challenges

KERIA Architecture and SSI Deployment Strategies

April 18, 2023

IIW 36

Agenda

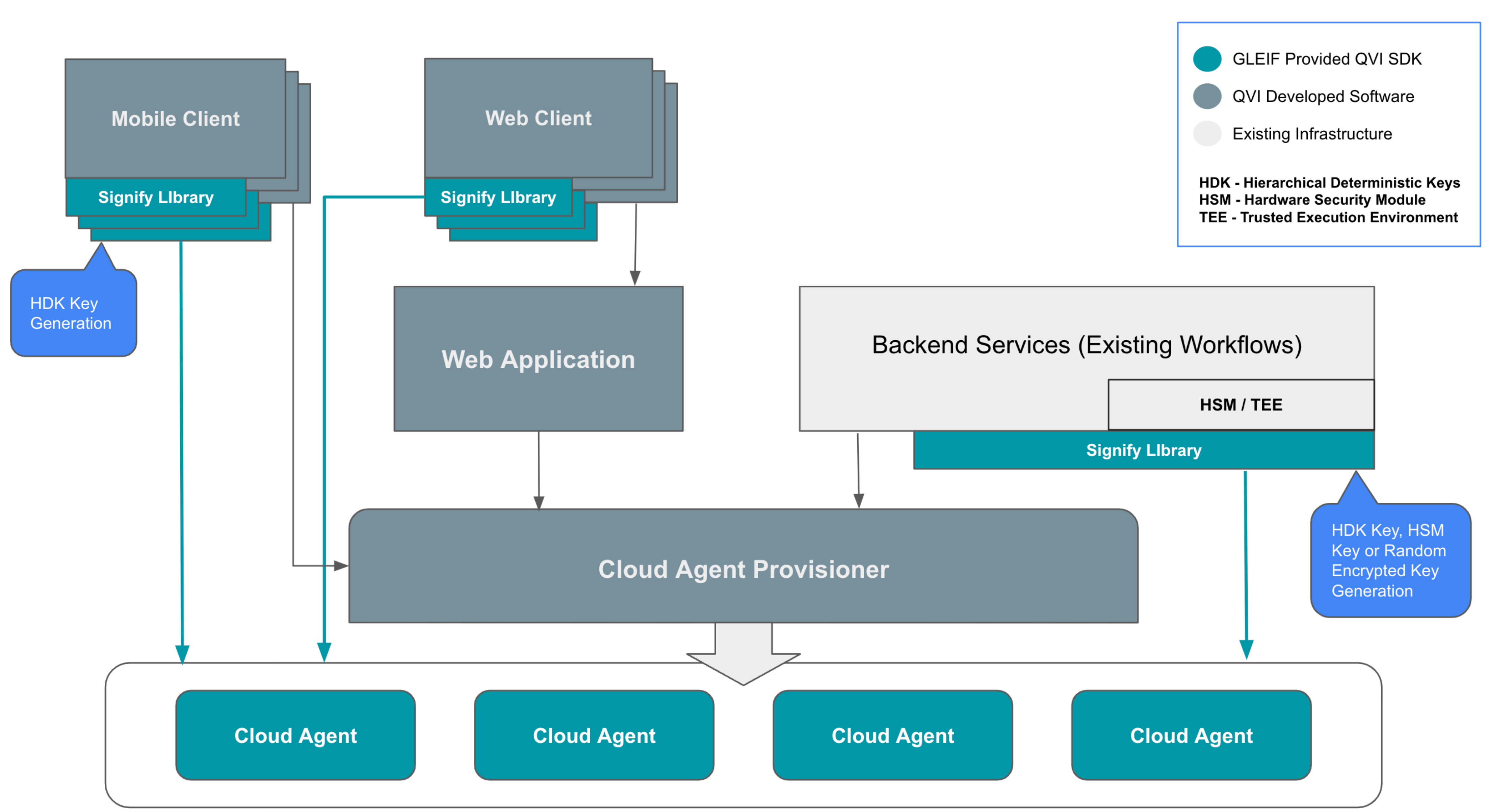
1. KERIA Overview
2. Signing at the Edge
   a) Salty Keys
   b) Randy Encrypted Keys
   c) Group Keys
   d) HSM / TEE Integration
3. Agent Role and OOBIs
4. Multi-Tenant Design
   a) Ports and Endpoints
   b) Configuration Options
5. Discussion
6. Plea for Help!

KERIA Overview

How we got here, where we're headed…

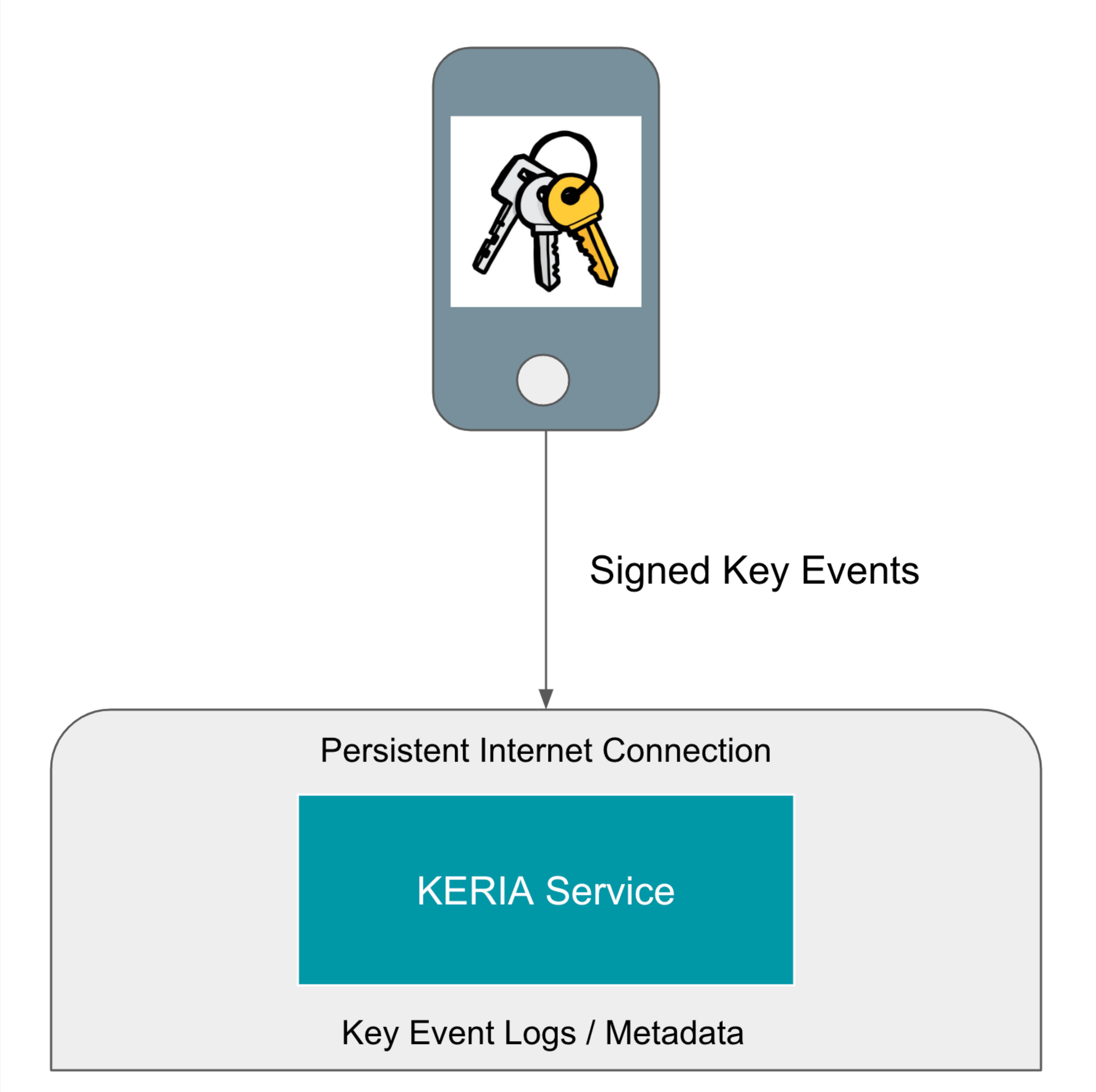
- **KERIpy** – KERI Core Library Reference Implementation
  - CESR Primitives
  - Event Parsing, Generation, Signing
  - Receipts and Witness Implementation
- **KLI** – KERI Command Line Interface
  - Staying put for now
- **KERIpy Agent (KIWI) and the KEEP** – Mark I Agent / Reference UI
  - Not Safe for Children
- **Signify** – Signing at the Edge Client with **Minimally Sufficient KERI**
  - SignifyPy – Signify Reference Implementation In Python
  - Signifide – Rust Implementation, Sits on CESRide
- **KERIA** – Mark II Agent
  - Supports Signify Clients
  - Multi-Tenant Design

GLEIF SDK Development Efforts



Signing at the Edge

Where are the keys?



Supported Key Types

- **Salty Keys – Hierarchical Deterministic Keychain**
  - 21 Character Passcode Entered by User
  - Stretch into Seed using Argon2id
  - Combined with a “path” to create Ed25519 hierarchical deterministic key chain
  - Passcode never leaves the client
  - Full key hierarchy can be regenerated if needed
  - Agent stores “path” and “tier”, Controller remembers passcode
  - No key material on Agent, generated on the fly every time

- **Randy Keys – Randomly generated, encrypted and stored on agent**
  - 21 Character Passcode Entered by User
  - Stretch into Seed using Argon2id
  - Ed25519 key pairs and X25519 encryption keys generated
  - All signing and rotation keys generated using random algorithm on client only
  - Private keys, next public keys encrypted on client and stored on Agent

Supported Key Types (continued)

- **Group AID – Distributed Multisig Group**
  - Multisig Group AID for Signify Client
  - Local AID from Signify from one of the other types

- **HSM / TEE Integration**
  - Apple Secure Enclave for example
  - Support for additional crypto algorithms are needed
    - secp256k1 for example

Agent Role and OOBIs

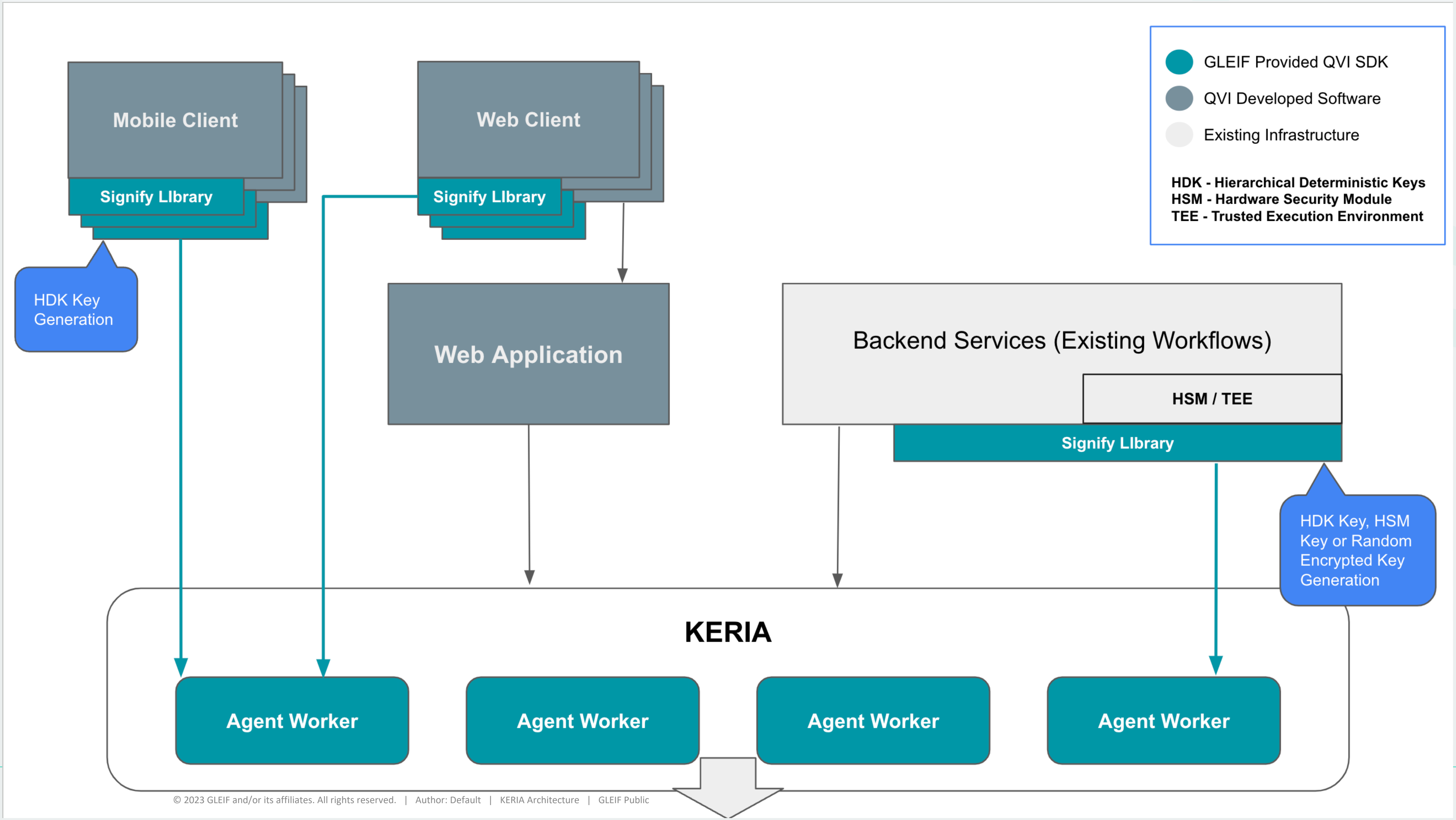
Agent Roles / OOBIS

- **New *agent* role added to supported roles**
  - Receives events directly to Agent
  - Provides persistent internet presence
  - Requires change to Multisig OOBI support.

- **New OOBI type for multisig group identifiers**
  - Multisig group communication must go to each member of the group
  - Endpoints must be extended to allow for groups of endpoints as a list, one for each member
  - OOBI exchange for a group will be an indexed list of endpoints associated with the members
  - Communication must be enhanced to support multicasting to group members
  - *http://example.com/oobi/<GROUP AID>/member/<INDEX>/<MEMBER AID>*

Multi-Tenant Design

Multi-tenancy



Mobile Client
Signify LIbrary
Web Client
Signify LIbrary
HDK Key Generation
Web Application
Backend Services (Existing Workflows)
HSM / TEE
Signify LIbrary
HDK Key, HSM Key or Random Encrypted Key Generation
KERIA
Agent Worker
Agent Worker
Agent Worker
Agent Worker
● GLEIF Provided QVI SDK
● QVI Developed Software
○ Existing Infrastructure
HDK - Hierarchical Deterministic Keys
HSM - Hardware Security Module
TEE - Trusted Execution Environment

Multi-tenancy (continued)

- **Ports and Endpoints**
  - 3 separate network interfaces exposed
    1. Boot Interface
    2. Agent Administrative Interface
    3. KERI Protocol Interface (to the rest of the world)

- **Configuration Options**
  - Network interfaces for each interface and port
  - Turn boot interface on or off
  - Preconfigured with existing (possibly static) agent/controller pairs
  - Typical KERI configuration providing witnesses, schema etc. to all Agents

Discussion

How Can I Help?

- **Repositories**
  - **KERIA** – http://github.com/WebOfTrust/KERIA
    - Python
    - Agent Service
    - Porting existing APIs, Docker, CI, Documentation, etc.
  - **SignifyPy** – http://github.com/WebOfTrust/signifypy
    - Python
    - Signify Client
    - Creating API client side classes, HSM integration, documentation, etc.
  - **Signifide** – http://github.com/WebOfTrust/signifide
    - Rust
    - Signify Client
    - FFI, WASM Bindings, API client side classes, CI, documentation, etc.

Plea for Help!

Limitations

- This presentation contains confidential and proprietary information and/or trade secrets of the Global Legal Entity Identifier Foundation (GLEIF) and/or its affiliates, and is not to be published, reproduced, copied, or disclosed without the express written consent of Global Legal Entity Identifier Foundation.
- Global Legal Entity Identifier Foundation, the Global Legal Entity Identifier Foundation logo are service marks of Global Legal Entity Identifier Foundation.