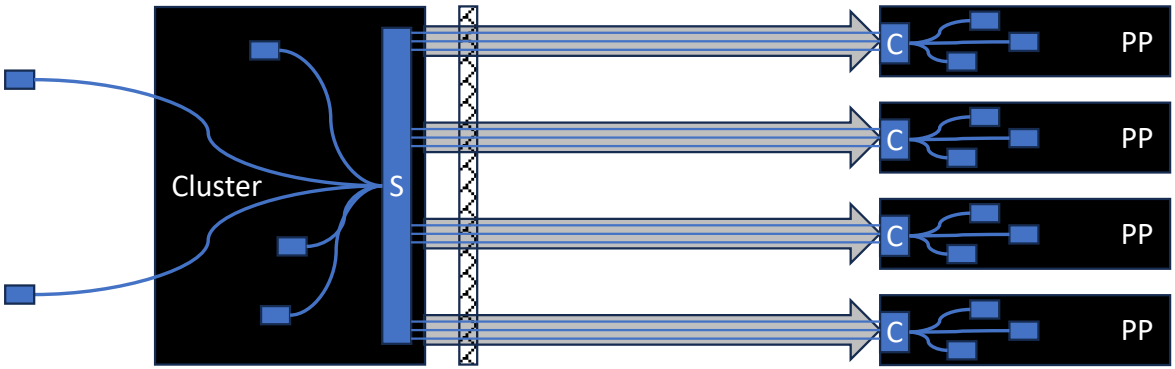
PP Secure Comms

- Enable serving PPs from a Cluster via firewall without exposing any required Cluster Services
- Enable securing any required communication between the Cluster and each PP using the SSH protocol
  - Resulting in a single channel between the Cluster and the PP
  - Any required tcp/udp communication is than handled using an SSH Tunnel
  - SSH Keys are secured through attestation
- Simplify PP configuration and provisioning - services use pre-fixed configuration
- Simplify Cluster configuration and provisioning - services use pre-fixed configuration

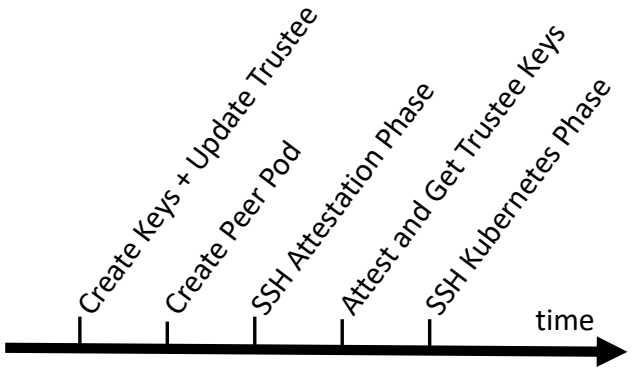


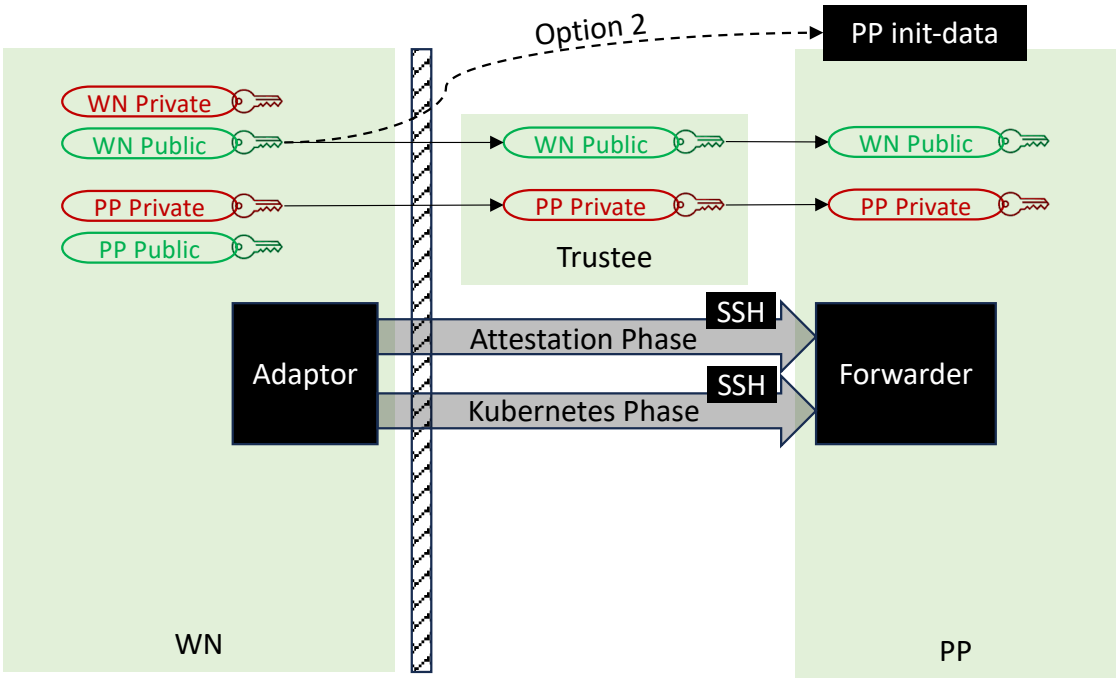
PP Secure Comms

Attestation Phase
Option 1: Use no keys
- Main concern: Hackers may race to hijack new PPs – Resolution: timeout hijacked PPs
- MITM feasiable but not a concern

Option 2: Use WN Public Key (I.e. a Public Key of the cluster worknodes)
- MITM and PP Hijacking is feasiable by cloud provider but not a concern

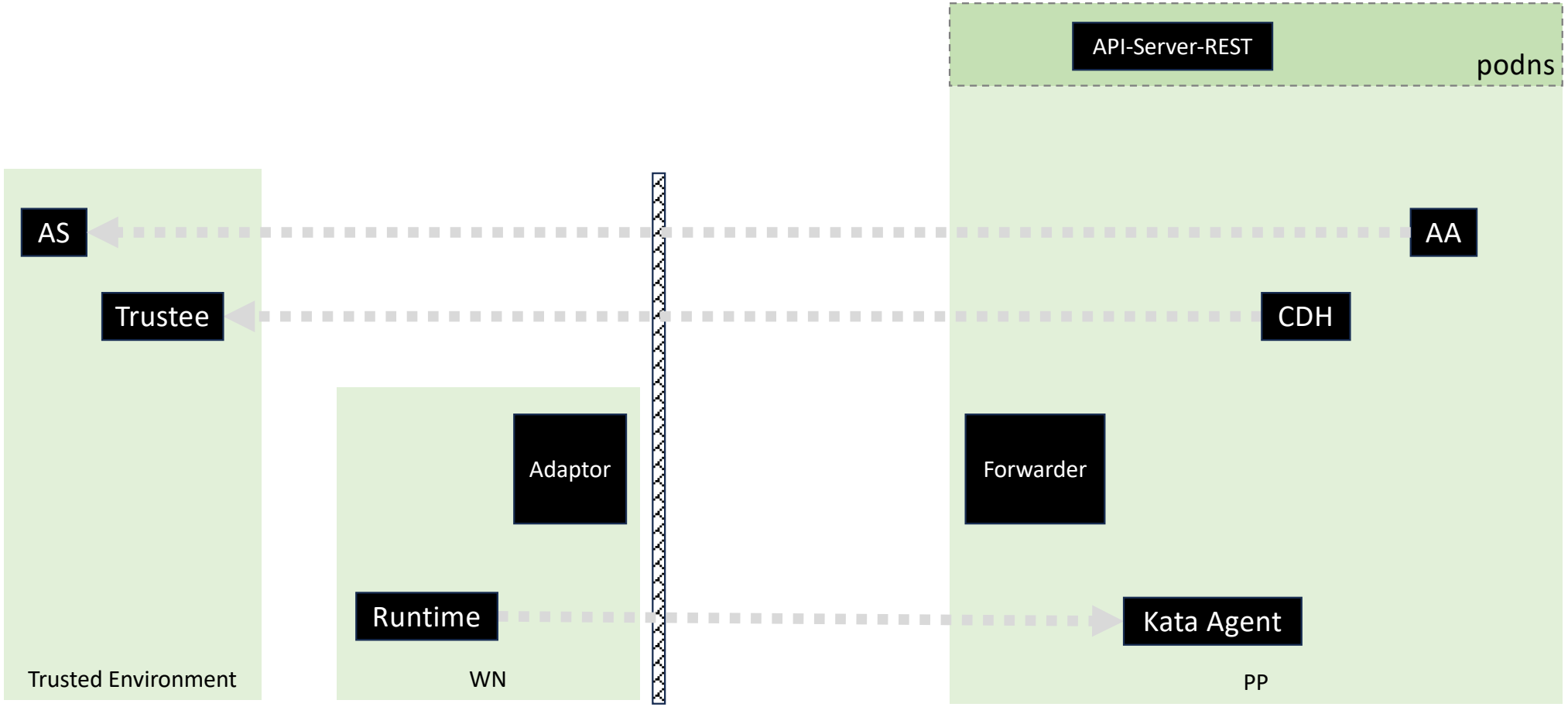
Kubernetes Phase
Use Keys from Trustee





Key Management:
- Adaptor creates WN Key Pair (shared by Cluster worknodes)
  - The WN Key Pair is reused for all PPs by all WNs
  - The WN Key Pair is kept as a Kubernetes Secret
  - The WN Public Key is sent to Trustee
  - The WN Public Key may be included in PP's init-data
- Adaptor creates PP Key Pair (per PP)
  - The PP Private Key is sent to Trustee
  - The PP Key Pair is kept as a Kubernetes Secret
  - The PP Key Pair is deleted when PP is destroyed
- Forwarder should ask Trustee for keys during **Attestation Phase**
  - When Keys are obtained Forwarder move to **Kubernetes Phase**
  - Forwarder is a singletone:
    - **Attestation Phase** allowed only once in the entire lifespan of the PP
    - A single SSH connection allowed for **Attestation Phase** (both concurent and consecutive)

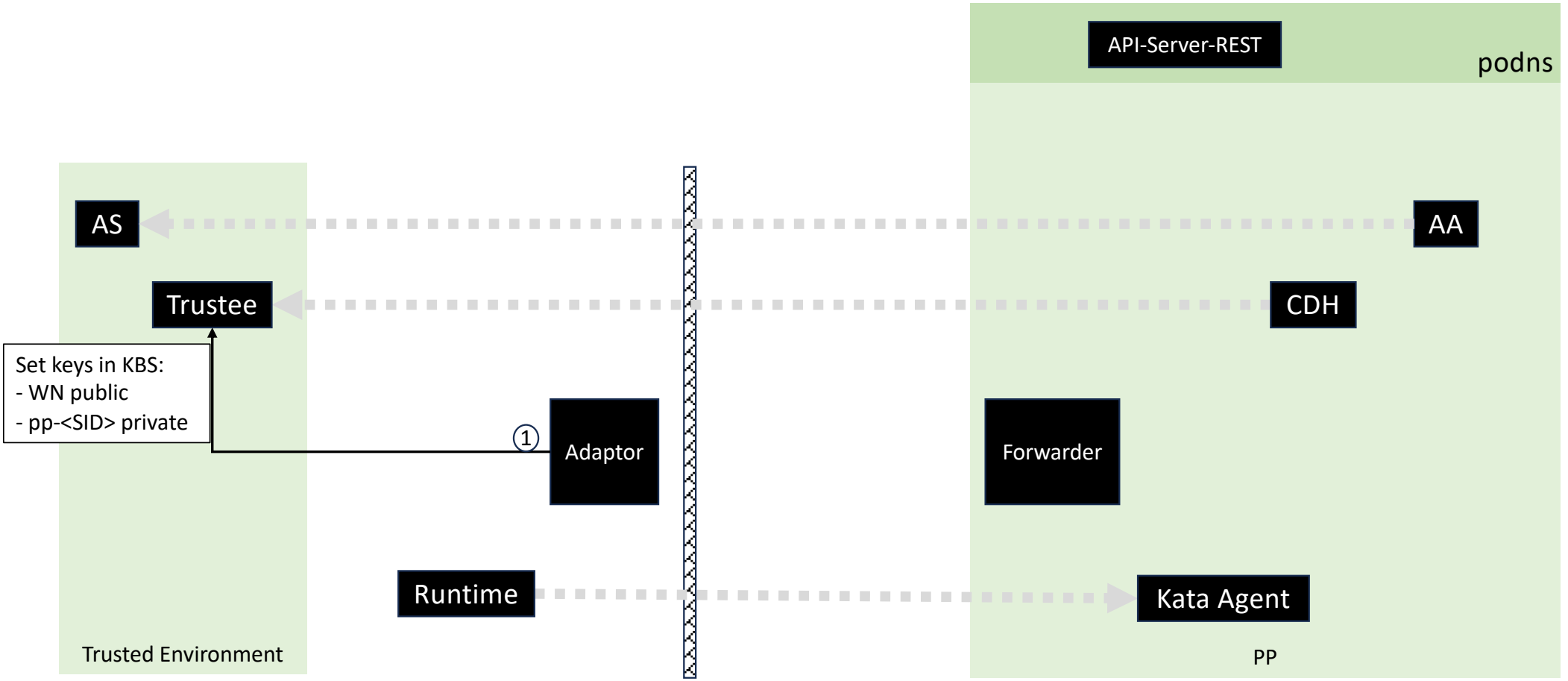
PP Secure Bootstrapping



PP Secure Bootstrapping

Initialization

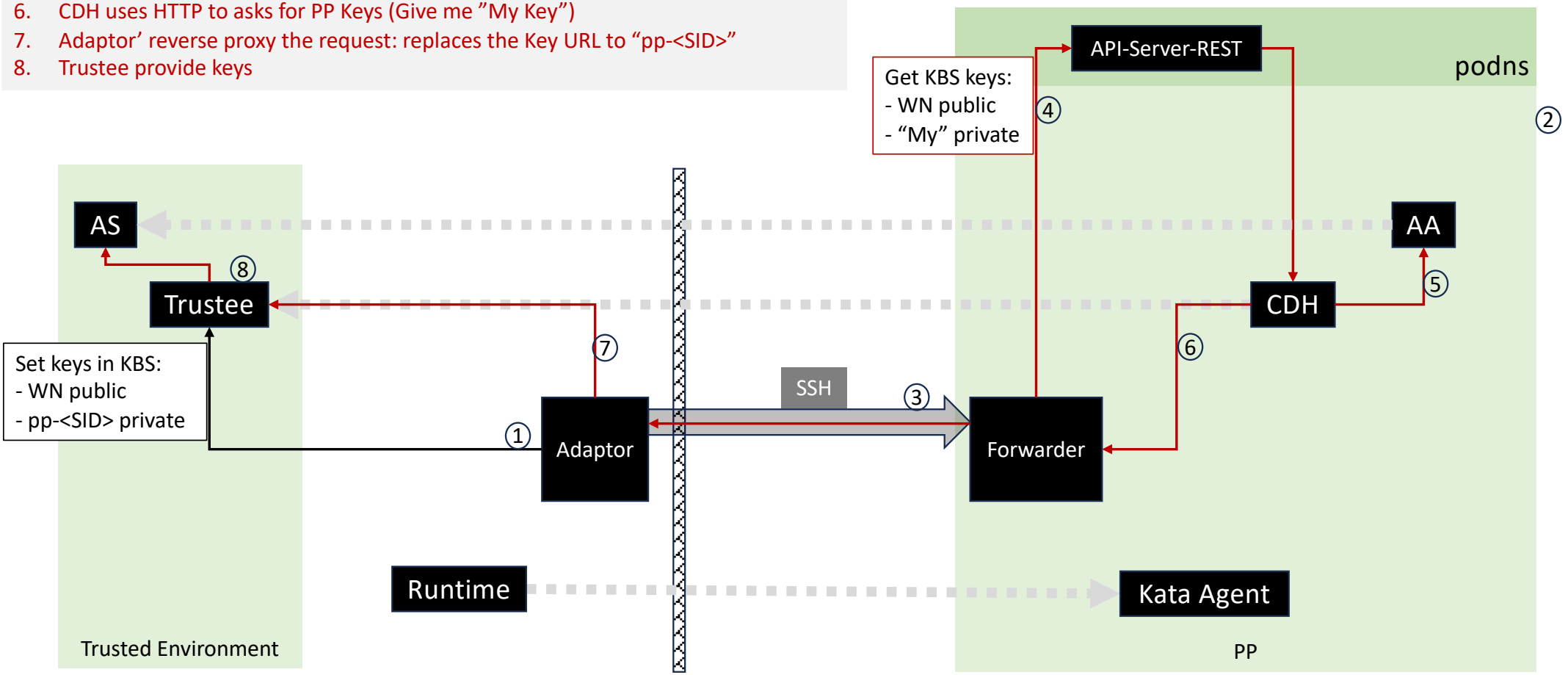
1. Adaptor sets keys for PP at Trustee



PP Secure Bootstrapping

Initialization | Attestation Phase

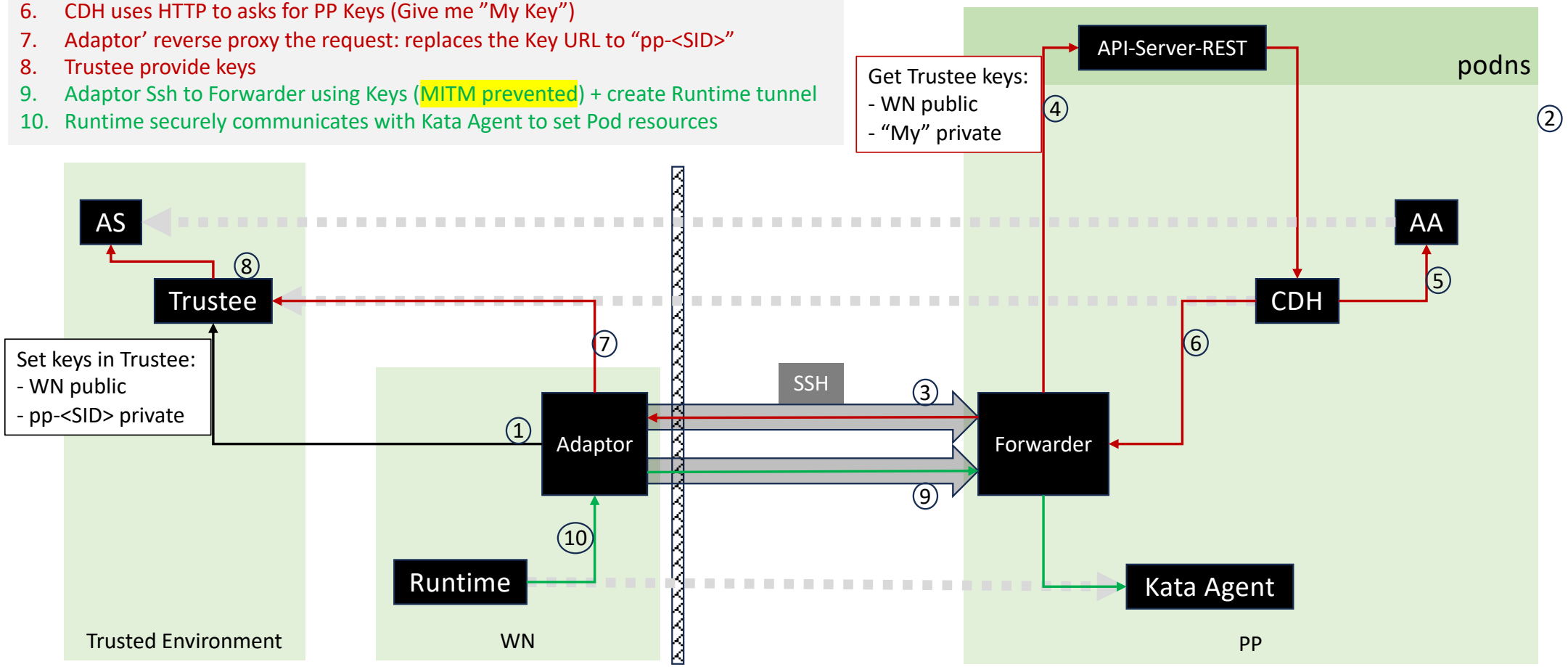
1. Adaptor sets keys for PP at Trustee
2. PP Created (Optional WN Public Key provided)
3. Adaptor Ssh to Forwarder + create CDH tunnel (MITM feasiable)
4. Forwarder asks for PP keys
5. Attestation
6. CDH uses HTTP to asks for PP Keys (Give me ”My Key”)
7. Adaptor’ reverse proxy the request: replaces the Key URL to “pp-<SID>”
8. Trustee provide keys



PP Secure Bootstrapping

Initialization | Attestation Phase | Kubernetes Phase

1. Adaptor sets keys for PP at Trustee
2. PP Created (Optional WN Public Key provided)
3. Adaptor Ssh to Forwarder + create CDH tunnel (MITM feasiable)
4. Forwarder asks for PP keys
5. Attestation
6. CDH uses HTTP to asks for PP Keys (Give me ”My Key”)
7. Adaptor’ reverse proxy the request: replaces the Key URL to “pp-<SID>”
8. Trustee provide keys
9. Adaptor Ssh to Forwarder using Keys (MITM prevented) + create Runtime tunnel
10. Runtime securely communicates with Kata Agent to set Pod resources



Assumptions, Concerns, Gaps

- Adaptor should delete the KBS Secret on VM Destory
- CDH must use HTTP
- Forwarder may ask CDH directly and not via API-Server-REST
- Adaptor Restarts are currently handled by restarting all PPs
  - Future work should include keeping a state on the PP to enable reconnecting PPs at the Kubernetes Phase.

PP Life Cycle

- If PP was started without WN Public Key, a random cloud attacker may connect before the WN
- If PP was started with WN Public Key, a hostile Public Provider may connect to the PP before the WN