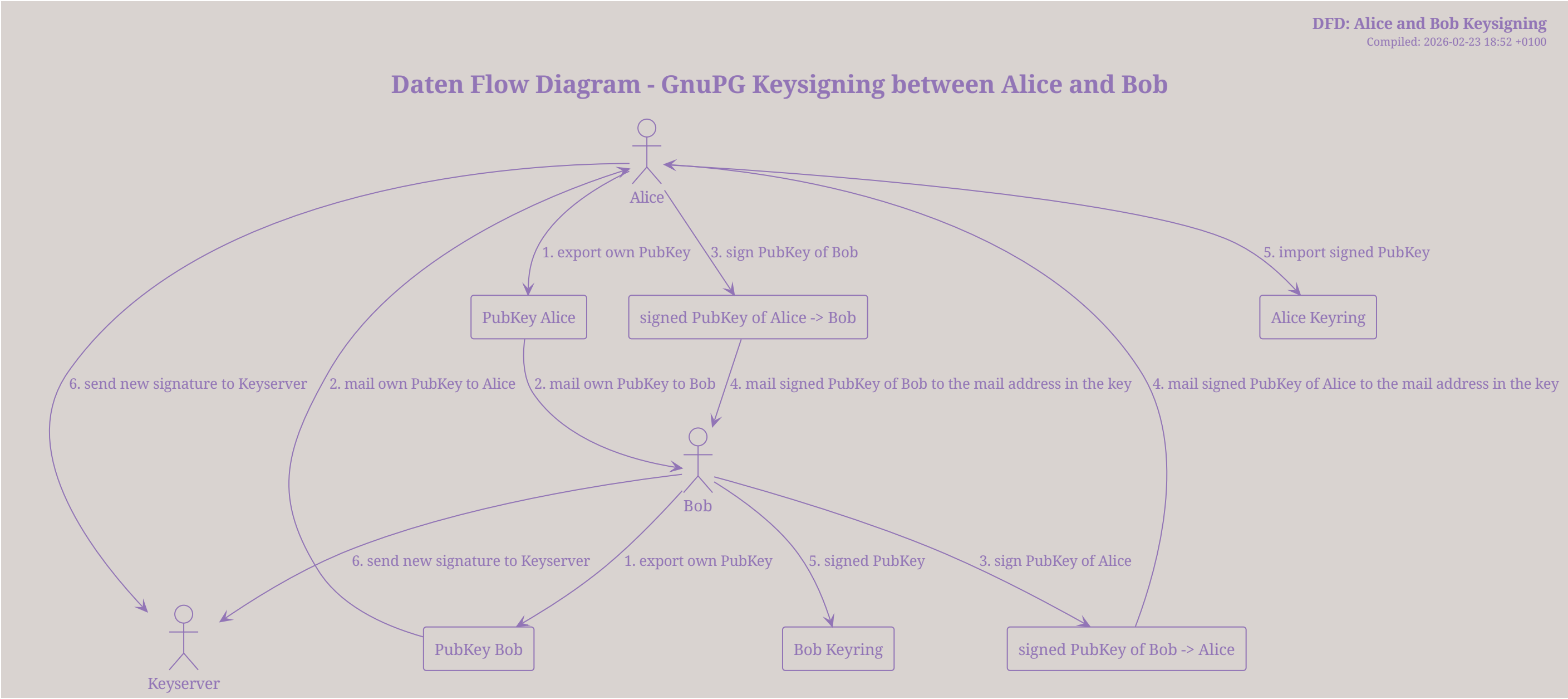


PLantUML Diagrams for GnuPG Key Signing

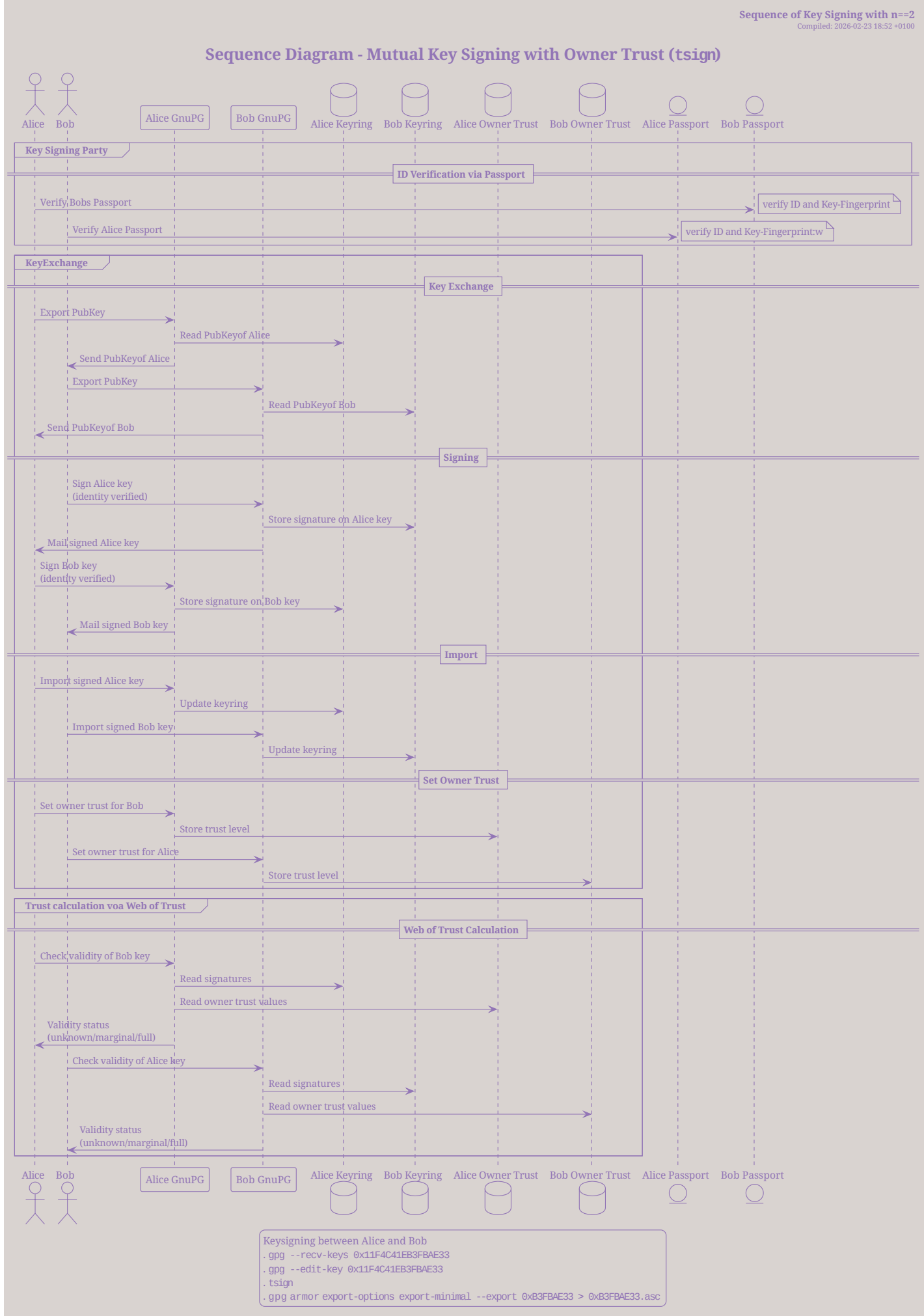
I am trying to model implicit and explicit trust in Zero Trust Architecture diagrams for Threat Modeling.

So I need to bring together the Blue Team/White Hat perspective and the Red Team/Black Hat stuff.

DFD: Keysigning Simple



SEQ: Keysigning with ID check and WoT



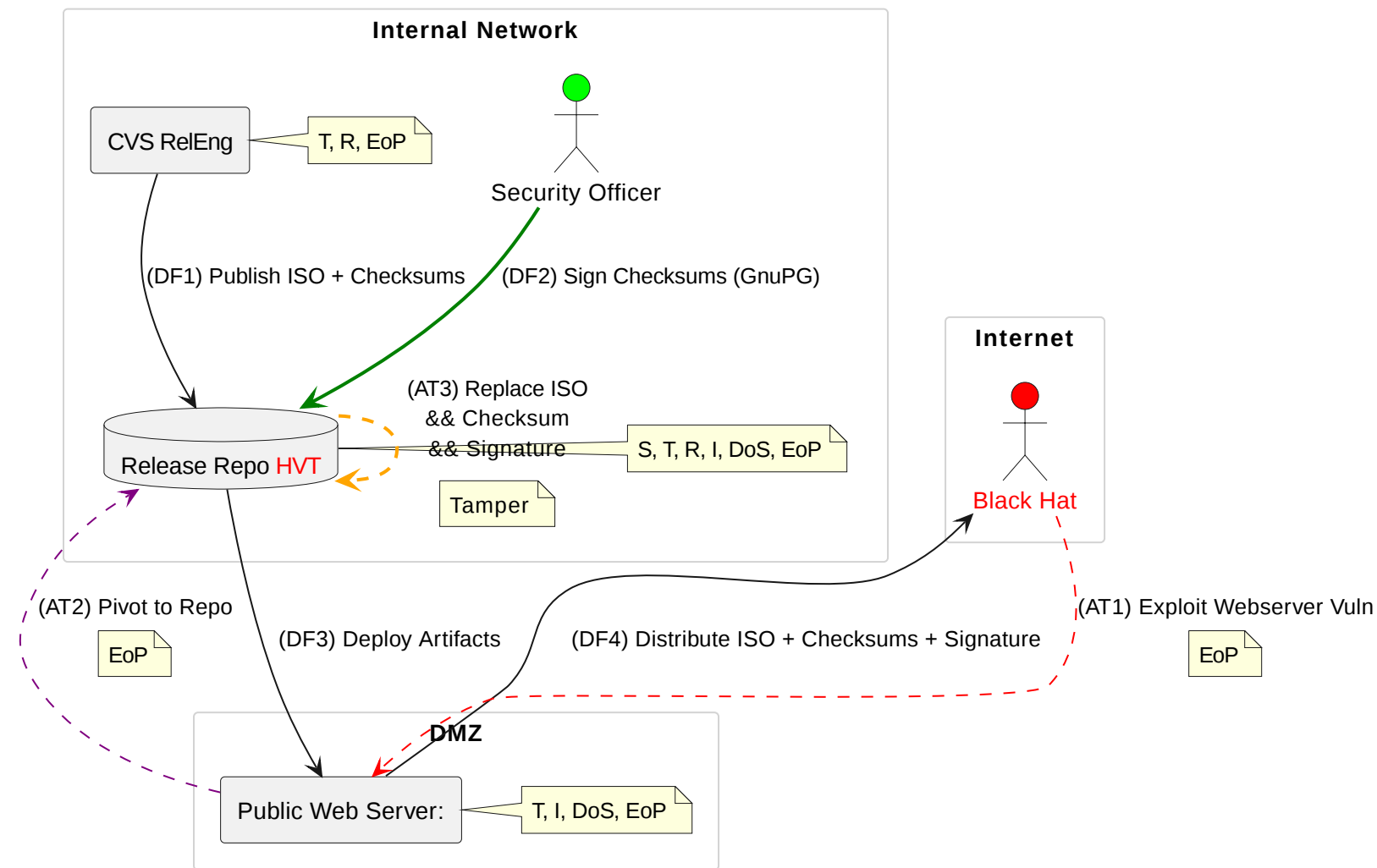
NetBSD RelEng Attack STRIDE

The ISO Image is built, signed and uploaded to the WWW server, as well as the Signature file and checksums.

Evil Black Hat hacks the webserver, and swaps the ISO image for a manipulated one with a valid Signature.

The manipulated Signature verifies the fake-integrity of the manipulated ISO image, but not the authenticity.

Attack Flow and Trust Boundaries: manipulated ISO image AND manipulated checksums/signatures (SLSA1)



STRIDE	Abbreviation
S	Spoofing
T	Tampering
R	Repudiation
I	Information Disclosure
DoS	Denial of Service
EoP	Elevation of Privilege

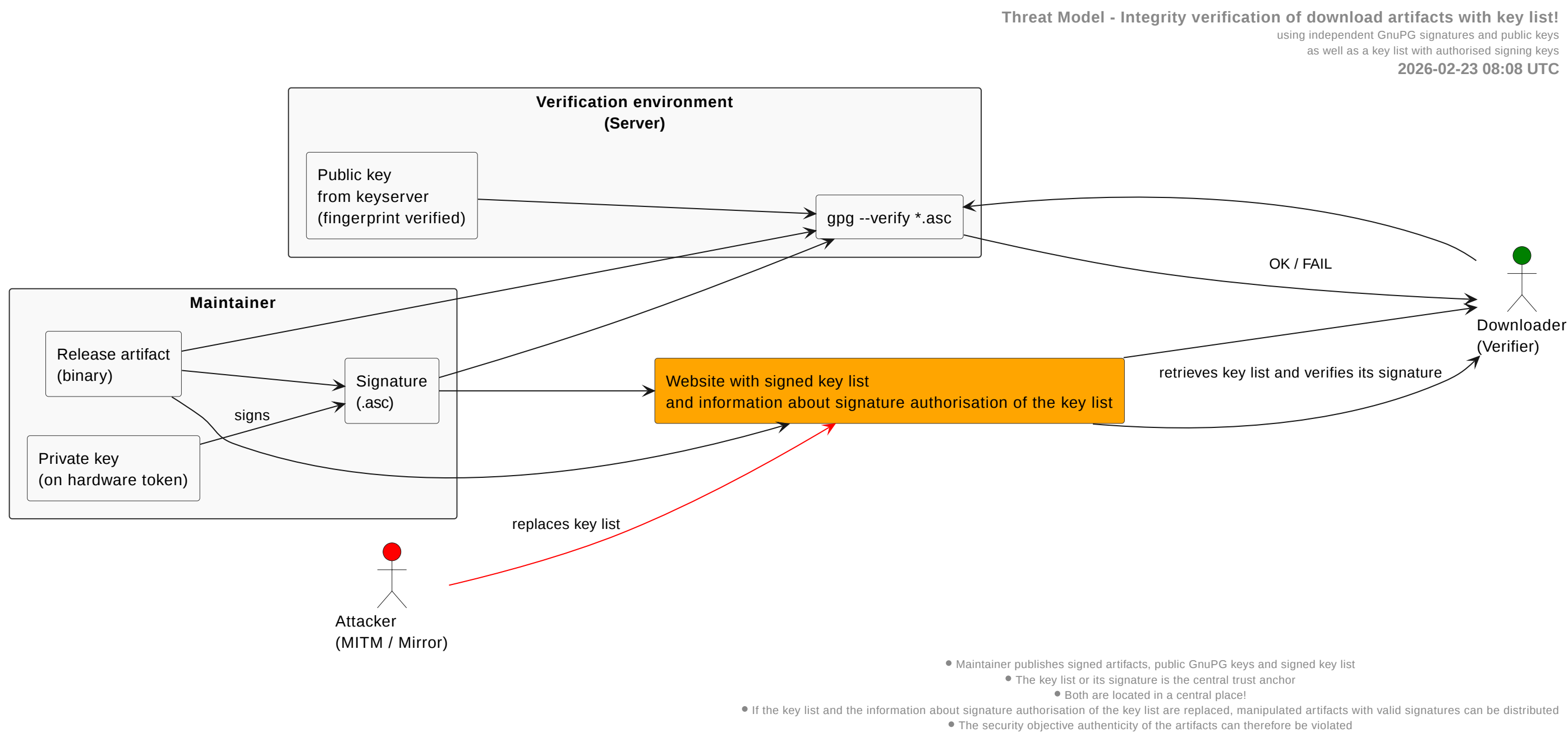
Relationship Colour

Colour	Attack Phase
Red	Initial Exploit
Deep Purple	Lateral Movement
Orange	Integrity Violation
Green	Legitimate Signing

LightGray box: Trust Boundary
Notes on Relation: STRIDE category
DF: Data Flow
AT: Attack Path

Supply-chain Levels for Software Artifacts

A simple threat model for SLSA Level 1 and Leve 4 build and distribution pipelines.

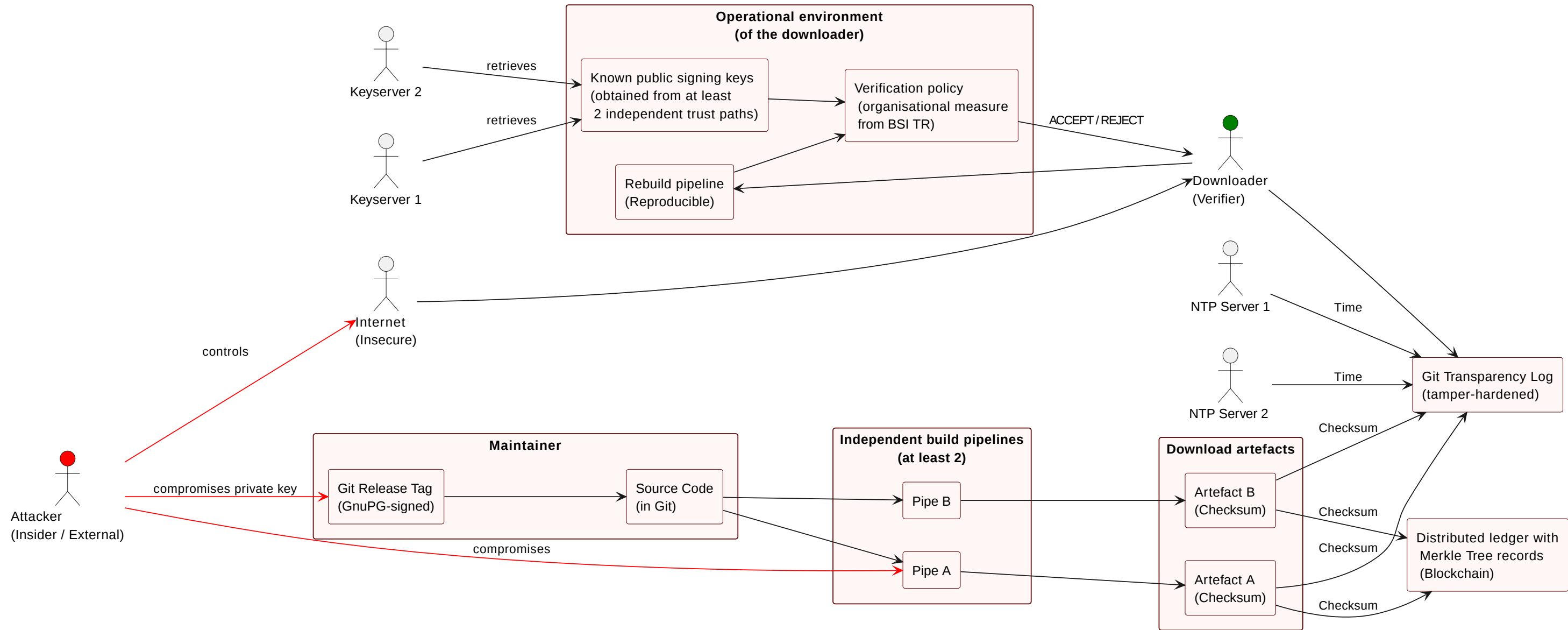


Reproducible Builds SLSA L4

Modeling Trust, Trust Anchors and Boundaries and Attack Vectors

The whole implementation

- Trust Boundaries: Rectangles
- Attack Vectors: Red Arrows

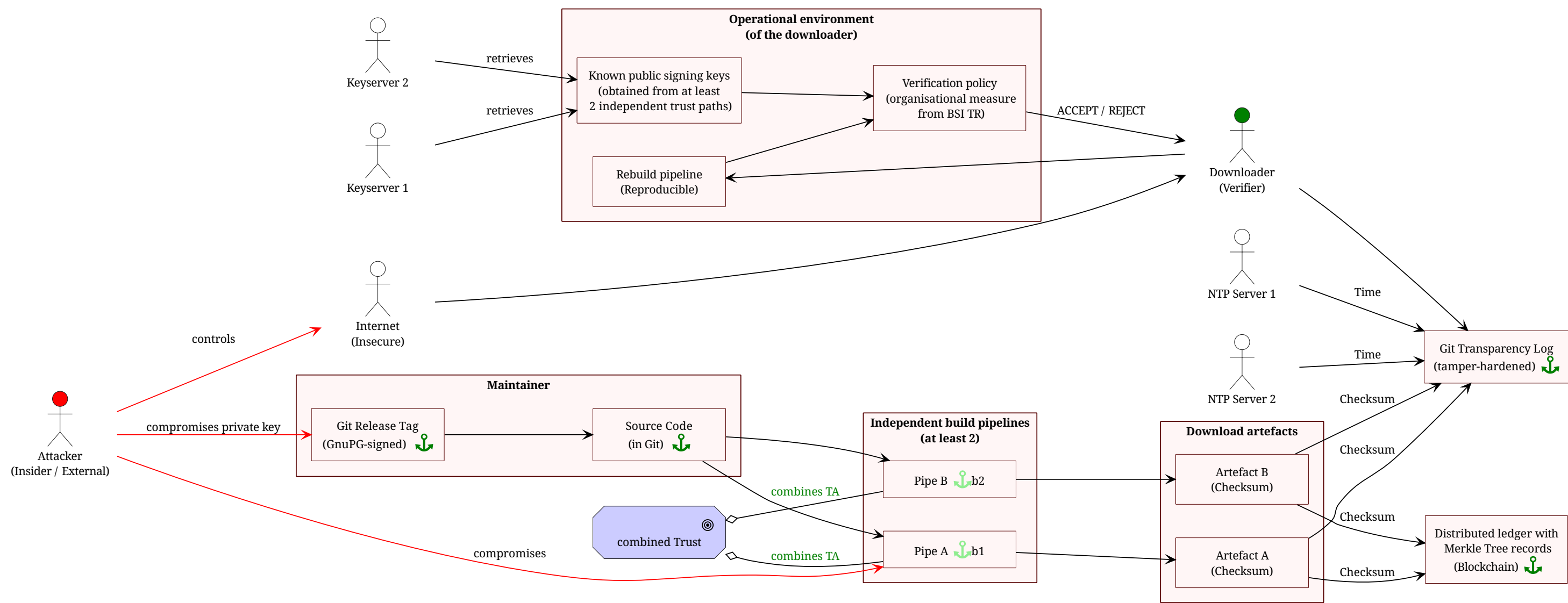


Threat Model - Reproducible Builds according to SLSA Level 4

Advantages:

- Consistency between source code and binaries through Reproducible Builds
 - No single trust anchor -> at least 2 independent builders
- Subsequent manipulation detectable through checksums in the Git Transparency Log
 - Insider attacks made more difficult through multiple signatures
- Key misuse detectable through checksums in the Git Transparency Log and temporal correlation via timestamps
 - Each box is a trust boarder

• Trust Anchor: Green Anchor



Threat Model - Reproducible Builds according to SLSA Level 4

Advantages:

- Consistency between source code and binaries through Reproducible Builds
 - No single trust anchor -> at least 2 independent builders
- Subsequent manipulation detectable through checksums in the Git Transparency Log
 - Insider attacks made more difficult through multiple signatures
- Key misuse detectable through checksums in the Git Transparency Log and temporal correlation via timestamps
 - Each box is a trust boarder

Motivation/Business Layer: Risk/Security Overlay

