Blockchain-based Secure Software Assets Management (BloSS@M)

Andrew Weiss, Lead Architect and Technical Advisor, UMBC

So what is the BloSS@M effort?

Demonstrate the feasibility and applicability of permissioned blockchain technologies in federal information systems

Leverage blockchain innovation and domain expertise shared between the University of Maryland, Baltimore County, DHS and NIST

Allow DHS as System Owner to demonstrate secure and compliant Blockchain architectures, system designs and automation of the A&A process to meet DHS and FISMA security and privacy requirements

Develop a proof of concept solution that highlights the use of blockchain for tracking federal software assets

Publish reference architectures and open source configuration artifacts and smart contracts

Blockchains and zero-trust

Traditional security centralized with “checks at the front door”

Zero trust “checks at every door in the building”

Blockchains decentralized mechanism for implementing a zero trust policy

Ledger immutability underpinned by the blockchain is a key driver of zero trust

Trust distribution in a blockchain and multi-cloud solutions

Blockchain trust model among distributed nodes with distinct governance <=> multi cloud trust model

Mutual trust is required   critical factor for new nodes joining the blockchain.

Rather than relying on a single, trusted intermediary (centralized), blockchains distribute trust via consensus mechanisms and public key cryptography

The “shared truth” is represented by the distributed ledger

Tracking government software assets with blockchain

Federal software asset acquisition and management is nontrivial and labor intensive

Blockchain is an ideal match to support the integrity, fairness and openness required of federal software procurement processes

- Acquisition and tracking activity types equate to transactions stored in the ledger (e.g. solicitations, awards, offers, software portfolio, etc)

Deduplicates software licenses and streamlines costs

Smart contracts are software that can be executed as transactions on the blockchain

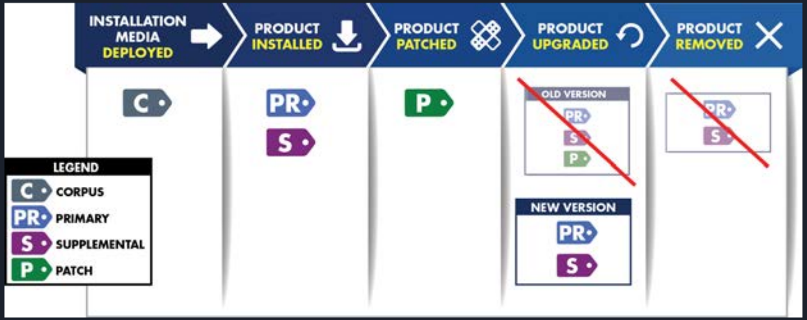
- Used to automate various software asset management functions (e.g. procurement event, license allocation, etc)

Putting it all together with BloSS@M

BloSS@M is a proof of concept **permissioned**, blockchain system that overlays a zero trust framework on a multi agency consortium

Smart contracts developed to ingest various data fields of software asset management activities

Integrated with the ISO/IEC Software Identification (SWID) tag format



Source

HYPERLEDGER FABRIC

Source

Bringing security control assessments to the blockchain

Core foundation for establishing the mutual trust among peer nodes

A core tenet of BloSS@M is to provide a path that agencies can follow to pursue an ATO for permissioned blockchain based systems

Blockchains require automated security control assessments due to the dynamic, distributed nature of the networks themselves

The Open Security Controls Assessment Language (OSCAL) is being leveraged by this effort

- Security content is formatted as OSCAL and stored in a separate system
- A separate blockchain network could theoretically be used to track the state of an OSCAL document itself (via hashing)

The technology behind BloSS@M

Leverage the Hyperledger ecosystem, which is an umbrella project of open source blockchain technologies hosted by the Linux Foundation

Built on the Hyperledger Fabric permissioned blockchain technology stack

- One of the most popular open source enterprise blockchain projects to date
- Adoption by major cloud service providers and large corporations around the world

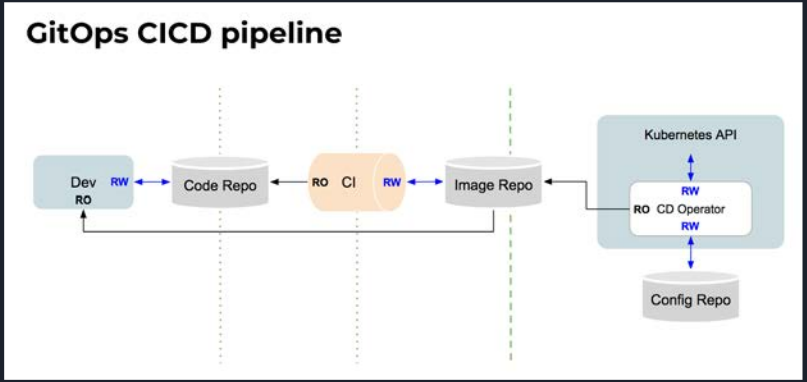
Smart contracts developed to demonstrate simplified government software asset tracking scenarios

BloSS@M - A DevSecOps Approach

Blockchain infrastructure built on "GitOps" workflows

Leverages the open source Hyperledger Labs Blockchain Automation Framework

- Deployment on Kubernetes based on DoD Enterprise DevSecOps reference architectures
- Built on Ansible, Helm and Flux



Source

BloSS@M - A DevSecOps Example

1. New peer organization requests to join and accepts policies/procedures outlined by the network’s governance framework
2. Peer organization provides new/existing Kubernetes cluster based on system requirements
3. Automated ATO process is initiated
4. Network configuration is updated with new organization’s details
5. ATO assessment of new infrastructure is complete
6. “GitOps” applies changes to the network based on updated configuration
7. SSP is automatically updated based on new configuration and results of assessment
8. Org is allowed to participate in endorsement of new transactions

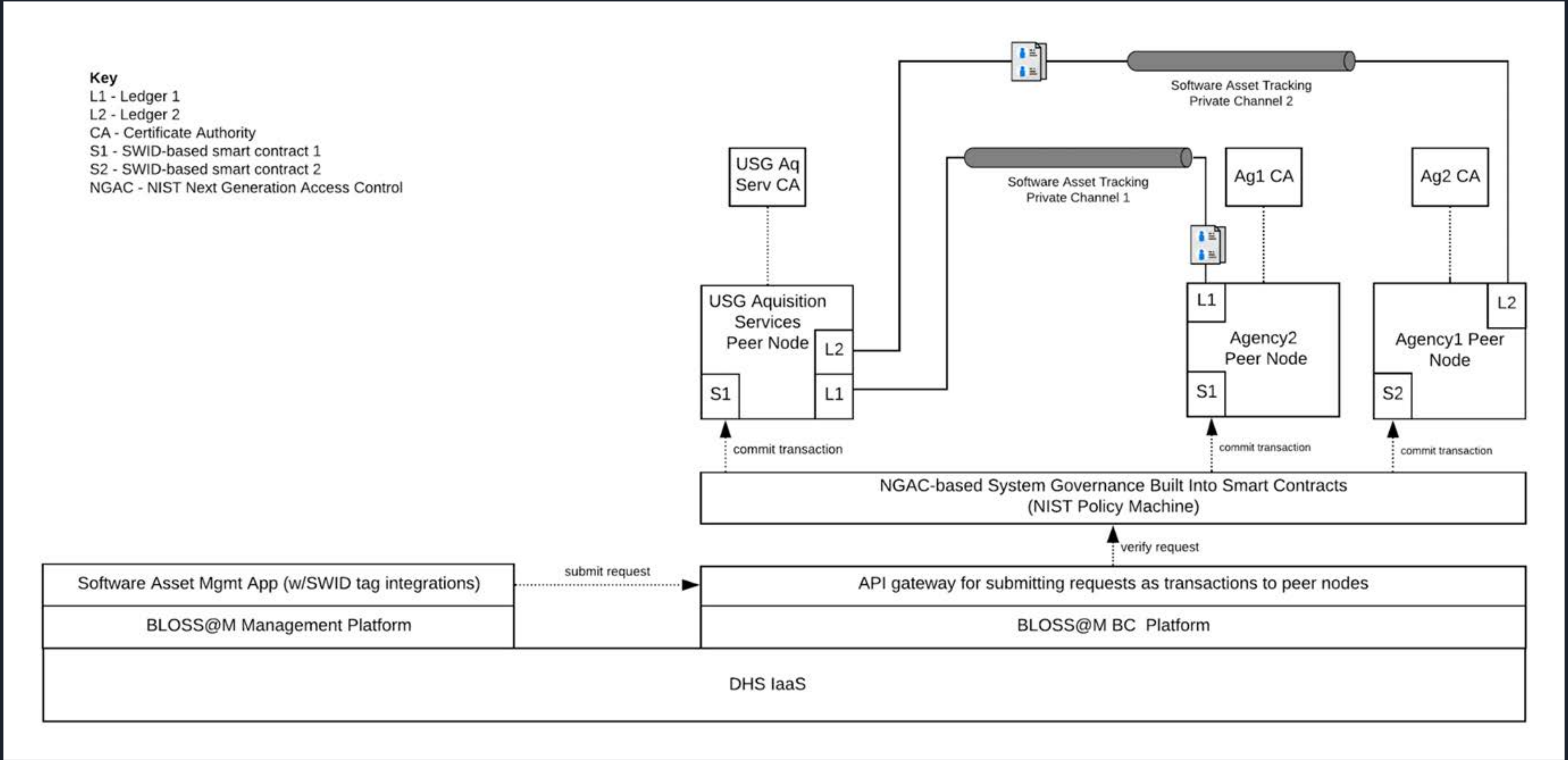
Diving further into the implementation

Smart contracts deployed to the blockchain network will enable:

Software asset management built on a generic JSON data model and the ISO/IEC Software Identification (SWID) tag format

Dynamic system ATO based on the Open Security Controls Assessment Language (OSCAL)

The network will employ fine grained access control capabilities provided by the NIST and ANSI/INCITS Next Generation Access Control framework (NGAC)


Key
L1 - Ledger 1
L2 - Ledger 2
CA - Certificate Authority
S1 - SWID-based smart contract 1
S2 - SWID-based smart contract 2
NGAC - NIST Next Generation Access Control
Software Asset Tracking Private Channel 2
USG Aq Serv CA
Software Asset Tracking Private Channel 1
Ag1 CA
Ag2 CA
USG Aquisition Services Peer Node
L2
L1
S1
L1
Agency2 Peer Node
S1
L2
Agency1 Peer Node
S2
commit transaction
commit transaction
commit transaction
NGAC-based System Governance Built Into Smart Contracts (NIST Policy Machine)
verify request
Software Asset Mgmt App (w/SWID tag integrations)
submit request
API gateway for submitting requests as transactions to peer nodes
BLOSS@M Management Platform
BLOSS@M BC Platform
DHS IaaS

BloSS@M Architecture

Participate

Open source development effort

https://github.com/usnistgov/blossom (coming soon)