NIST Internal Report
NIST IR 8500A ipd

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

Initial Public Draft

Michaela Iorga
Ned Goren
Dmitry Cousin
Selena Xiao
Marilyn Nguyen
Joshua Roberts

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8500A.ipd

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Internal Report
NIST IR 8500A ipd

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

Initial Public Draft

Michaela Iorga
Ned Goren
Dmitry Cousin
Selena Xiao
Marilyn Nguyen
Joshua Roberts
Computer Security Division
Information Technology Laboratory

This publication is available free of chargfrom:
https://doi.org/10.6028/NIST.IR.8500A.ipd

May 2026

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting Under Secretary of Commerce for Standards and Technology and Acting NIST Director

Certain commercial equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

NIST Technical Series Policies
Copyright, Use, and Licensing Statements
NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on YYYY-MM-DD [Will be added in final publication]

How to Cite this NIST Technical Series Publication:
Iorga M, Goren N, Cousin D, Xiao S, Nguyen M, Roberts J (2026) Blockchain-Based Secure Software Assets Management (BloSS@M). (National Institute of Standards and Technology, Gaithersburg, MD), NIST Internal Report (IR) NIST IR 8500A ipd. https://doi.org/10.6028/NIST.IR.8500A.ipd

Author ORCID iDs
Michaela Iorga: 0000-0001-7880-6045
Ned Goren: 0009-0009-3578-2958
Dmitry Cousin: 0000-0001-7148-4039
Selena Xiao: 0009-0001-2991-3175
Marilyn Nguyen: 0009-0004-1186-5613
Joshua Roberts: 0000-0001-8955-1375

Public Comment Period
May 19, 2026 – June 26, 2026

Submit Comments
blossom@nist.gov
National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Additional Information

Additional information about this publication is available at https://csrc.nist.gov/pubs/ir/8500/a/ipd, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

The report proposes a conceptual aggregation model for software acquisitions. The proposed approach relies on the immutability and auditability of blockchain technology. The model also enables automated, dynamic queries to the National Vulnerability Database (NVD) to continuously identify newly disclosed vulnerabilities associated with leased software assets. In parallel, the digitization of asset-level security and compliance information using the Open Security Controls Assessment Language (OSCAL) supports machine-readable and tool-consumable risk assessment, continuous monitoring, and life cycle-based risk management workflows. This proposed approach could be utilized for federal software acquisition to enable interagency sharing, reuse, and lifecycle management of software assets. It also has the potential to significantly increase collective purchasing power, reduce duplicative procurements, and strengthen supply chain security and IT asset management practices.

Keywords

acquisition; assessment; authorization-to-operate; blockchain; controls; open security controls assessment language; OSCAL; risk management; security; software supply chain; vulnerability.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems.

Call for Patent Claims

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

   i. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or

   ii. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: blossom@nist.gov

Table of Contents

List of Figures

1. Introduction

Information technology (IT) and operational technology (OT) [1] systems rely on complex, globally distributed supply chains to deliver software-enabled capabilities. These supply chains involve multiple organizations, distributed networks, technologies, and regulatory frameworks that collectively support the life cycle of IT/OT products and services, including design, development, integration, deployment, operation, and maintenance.

The distributed and heterogeneous nature of IT/OT supply chains presents challenges in establishing and maintaining visibility into the provenance and life cycle status of IT/OT assets. Within a single organization, it may be difficult to determine the origin of an asset, the testing or certification that the asset has undergone, the entities that have handled it, and its current operational status or location. Limited or inconsistent inventory visibility can impede an organization's ability to manage cybersecurity risks and implement effective asset governance practices.

These challenges are pertinent to any assets (e.g., hardware, firmware, software) but are particularly pronounced for software assets. Software asset management encompasses the processes used to track and manage software acquisition, deployment, configuration, maintenance, update, and retirement activities. Accurate and current asset inventories are necessary to support configuration control, access enforcement, life cycle planning, and cost accountability. Patch management is an additional burden for all software and firmware assets.

Federal departments and agencies are subject to asset inventory and management requirements established by Office of Management and Budget (OMB) Circular A-130 [2], OMB Memorandum M-13-13 [3], NIST Special Publication (SP) 800-37 [4], and SP 800-53 [5]. Despite these requirements, organizations frequently report incomplete, inconsistent, or outdated inventories of IT assets. Contributing factors include decentralized organizational structures, the use of disparate asset tracking and labeling systems, workforce turnover, rapid asset movement, and ongoing system modernization and re-engineering. Although automated discovery and management tools have improved asset visibility, challenges remain in achieving accurate, authoritative, and interoperable software asset inventories across organizational boundaries.

To help address these challenges, this report introduces *Blockchain-based Secure Software Asset Management (BloSS@M)*. BloSS@M leverages blockchain technology to support the creation and maintenance of a shared, tamper-resistant record of software asset life cycle information. The approach is intended to improve inventory integrity, support verifiable life cycle tracking, and enable consistent software asset management across diverse IT/OT environments. BloSS@M is a conceptual approach for software assets, but it can easily be expanded to firmware and hardware assets. Subsequent sections of this report describe the BloSS@M concept, its design considerations, and its applicability to federal software asset management use cases.

2. BloSS@M: The Proposed Solution

The proposed solution focuses on a coordinated approach to managing software assets across U.S. Government agencies rather than relying on independent, agency-specific procurement and license management processes. The BloSS@M model explores a software leasing service in which blockchain technology is used to record software asset life cycle information and to support the controlled distribution of software usage rights to authorized users. Under this model, software licenses are treated as shared, managed assets that can be provisioned and reassigned as operational needs change. This approach is intended to improve flexibility and support more efficient government-wide use of software resources. Blockchain technology is used to maintain an immutable and auditable record of software license agreements, allocations, and usage events, thus enabling participating organizations to verify asset status and usage without relying on a single centralized authority (see Sec. 3.1 for information on the blockchain technology pertinent to the current document).

The BloSS@M model is intended to address long-standing challenges associated with fragmented software procurement, cost, and license management, including limited visibility into software acquisition and usage across agencies. By supporting shared visibility and coordinated management, the proposed solution seeks to improve asset accountability, enable more informed contract negotiations, and enhance software supply chain risk management. Blockchain technology is used to support these objectives in environments where participating departments, agencies, vendors, and publishers may not fully trust one another.

For the purposes of this model, it is assumed that software acquisition and licensing are managed centrally on behalf of the U.S. Government, while individual agencies and departments act as authorized users of those managed software assets. This assumption aligns with the role of the Quality Service Management Organization (QSMO) [6], as described in the President's Management Agenda (PMA), which promotes the consolidation of common administrative services to improve efficiency, transparency, and cost management. The proposed solution is also designed to support federal cybersecurity priorities emphasized in recent Executive Orders, including strengthening software supply chain security, improving visibility into third-party software risks, and increasing the use of automation to support continuous risk management.

BloSS@M enables the comparative evaluation of software assets that provide similar functionality across agencies by integrating cost, licensing, and security-relevant attributes into a unified management framework. Security posture information is derived from Open Security Controls Assessment Language (OSCAL) [7][8] Component Definition artifacts [9] that describe implemented security controls and configuration characteristics in a machine-readable format. The service also supports automated correlation with authoritative external data sources, including the National Vulnerability Database (NVD) [10] for known vulnerabilities and the MITRE ATT&CK [11] and MITRE D3FEND [12] frameworks for threat and mitigation mapping. These integrations enable automated and AI-assisted (including agentic and generative) provisional assessments that evaluate how implemented controls align with known adversarial techniques and defensive measures. Collectively, these capabilities are intended to support

secure-by-design principles, improve supply chain risk-informed decision-making, and enable scalable, evidence-based software governance across the federal enterprise.

A graphical representation of BloSS@M is provided in Fig. 1. The detailed description of the represented entities will be discussed in the sections that follow.

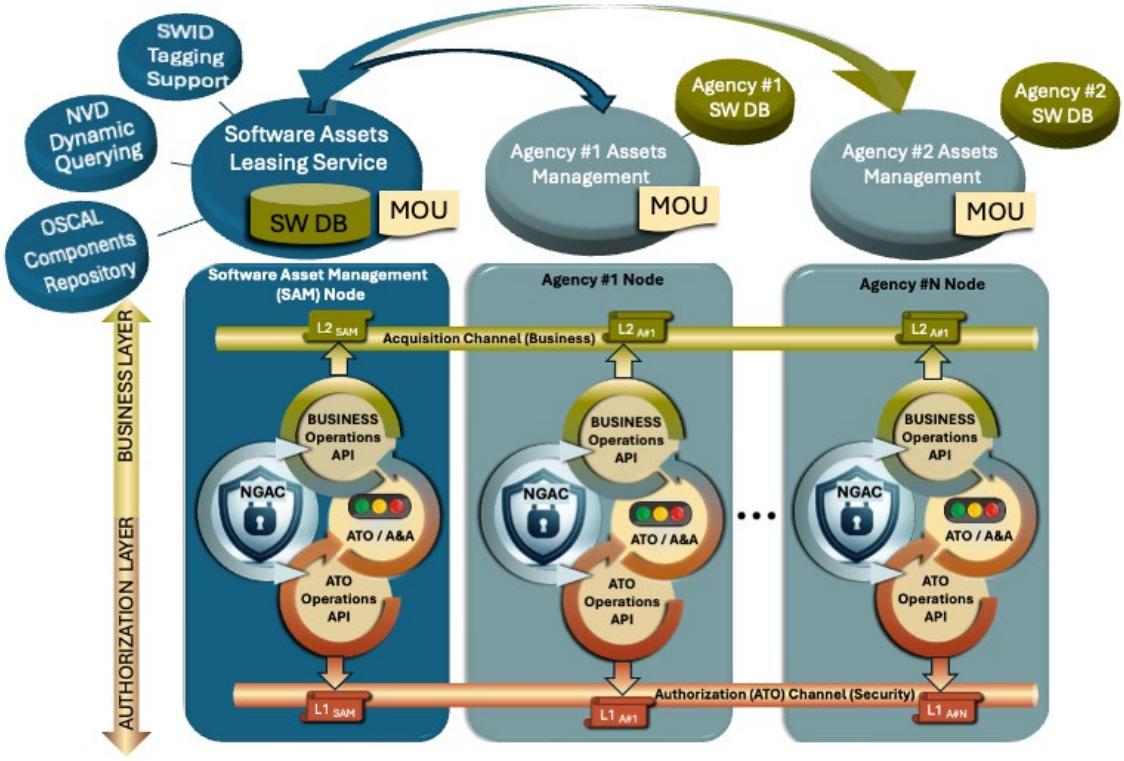


Fig. 1. BloSS@M overview

The following subsection describes the core capabilities of the BloSS@M model in greater detail. These capabilities illustrate how BloSS@M supports software asset comparison, life cycle visibility, security posture assessment, and automated risk analysis by leveraging standardized representations, authoritative data sources, and advanced analytical techniques. Each capability is described in terms of its purpose, inputs, and contribution to federal software asset management and supply chain risk management objectives. Leveraged technologies that enable these capabilities are explained in greater detail in Sec. 3.

2.1. BloSS@M Capabilities

BloSS@M provides a set of operational and technical capabilities that collectively support coordinated software asset management, supply chain risk visibility, and automation-enabled cybersecurity governance. Collectively, these capabilities can be logically grouped into a *BloSS@M business layer*. The capabilities are designed to improve transparency, comparability, and security awareness across the software asset life cycle while supporting enterprise-wide decision-making. The service's infrastructure and operational capabilities and processes are

logically grouped and referred to as the *BloSS@M operational layer*. This layer also governs the members' access to the *BloSS@M business layer (*see Sec. 6 for more details).

Figure 2 isolates the business layer of the BloSS@M system to focus on offered operational and technical capabilities.

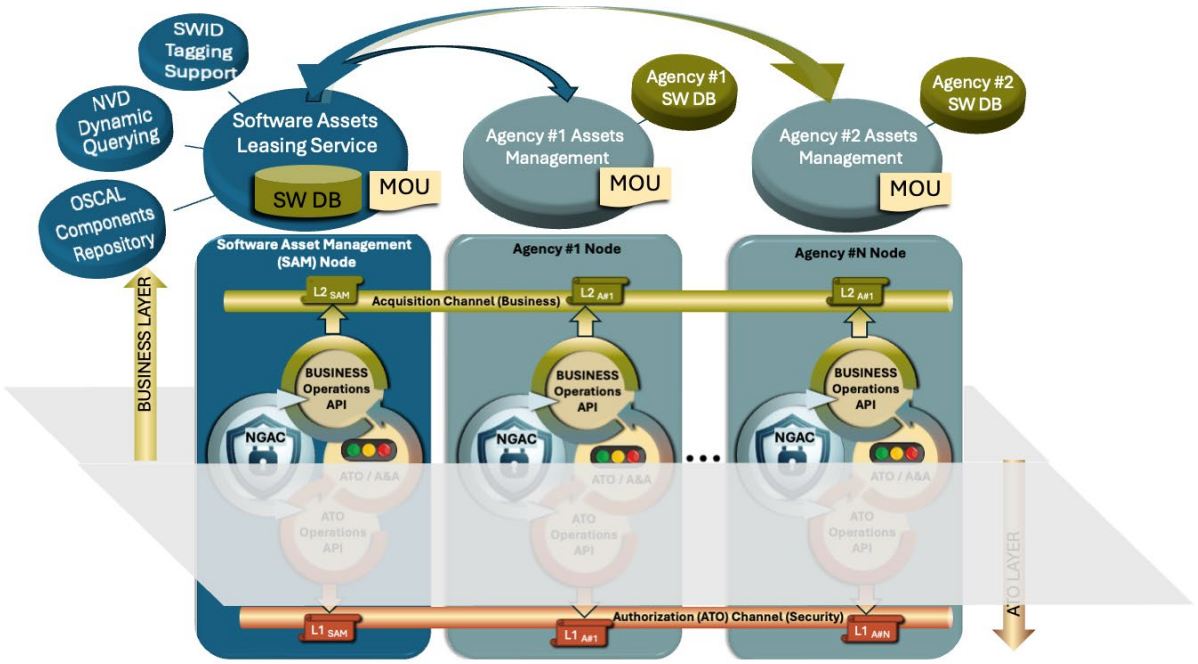


Fig. 2. BloSS@M overview of the business layer

2.1.1. Operational Capabilities

2.1.1.1. Asset Discovery

BloSS@M supports the enterprise-level discovery and visibility of software assets that are available for use across participating organizations. The model enables the identification of software products based on defined functional, contractual, or policy-based criteria to provide users with a consolidated view of available software assets. This capability supports inventory accuracy and reduces fragmentation caused by decentralized procurement and asset tracking practices.

Asset discovery is a capability that could be extended to firmware and hardware assets acquisition should an entity build and operate a centralized, government-wide, BloSS@M-like non-software assets acquisition service.

2.1.1.2. Asset License Cost Visibility

BloSS@M maintains visibility into software license or leasing cost information as provided by software publishers or vendors. Cost data is associated with each software asset and made available to authorized users to support budgeting, life cycle cost analysis, and procurement planning. This capability enables consistent cost comparison across similar software offerings and supports enterprise-level financial oversight.

For a hardware and firmware assets acquisition service, an *Asset Cost Visibility* could be considered. The capability would also enable cost comparison across similar hardware or firmware offerings and supports enterprise-level financial oversight.

2.1.1.3. Asset Comparison

BloSS@M enables the comparative analysis of software assets that provide similar functionality. Comparison criteria may include functional characteristics, licensing or leasing terms, cost, and security-relevant attributes. By normalizing this information across products and vendors, BloSS@M supports informed decision-making and promotes the reuse of approved software assets across agencies.

Asset Comparison is a capability that could also be extended to a centralized, government-wide firmware and hardware acquisition service.

2.1.1.4. Asset Leasing and Traceability

BloSS@M supports a software leasing model in which usage rights are centrally managed and dynamically allocated to authorized users. This capability enables the reassignment and optimization of software license usage based on operational demand, thus improving utilization and reducing redundant acquisitions. Leasing activity and usage events are recorded to support accountability and auditability.

The *asset leasing and traceability* capability supports end-to-end software asset life cycle management by leveraging ISO/IEC 19770-2 *software identification (SWID) tags* in conjunction with defined operational and governance processes to maintain authoritative visibility into software usage and states over time. SWID tags provide standardized, machine-readable metadata that uniquely identify software products, versions, editions, patch levels, and relationships (e.g., installation, update, and removal events). BloSS@M uses this information to continuously monitor license entitlement and consumption, track installation and uninstallation activities across environments, and verify the application of updates and security patches throughout the asset life cycle.

2.1.2. Technical Capabilities

2.1.2.1. Asset Secure Configuration and Control Implementation

BloSS@M integrates security configuration and control implementation information using OSCAL Component Definition artifacts. These artifacts provide machine-readable descriptions of implemented security controls and related characteristics for software assets. This capability enables consistent representation and allows for the automated analysis of a software security posture across products, platforms, and vendors.

Using the OSCAL Component Definition model, vendors can create artifacts that document cryptographic elements, provenance, implemented security controls, and known vulnerabilities for any software, firmware, or hardware component. These OSCAL artifacts can then be linked to any chosen Bill of Material (BOM) using OSCAL's built-in extension mechanisms [7].

In the context of a BloSS@M-like service for hardware and firmware assets acquisition, OSCAL Component Definition artifacts can also incorporate pointers to application-specific integrated circuit (ASIC) documentation.

2.1.2.2. Asset Vulnerability Identification

BloSS@M supports the continuous association of software assets with known vulnerabilities by integrating with the NVD [10]. Vulnerability information includes affected products, vulnerability identifiers, and impact metrics. This capability enables organizations to assess exposure, prioritize remediation, and factor vulnerability considerations into software selection and life cycle management decisions.

Asset Vulnerability Identification is a capability that could also be extended to a centralized, government-wide firmware and hardware acquisition service.

2.1.2.3. Asset Threat Overlay

BloSS@M supports threat analysis by correlating software asset security characteristics with the known adversarial techniques and behaviors documented in the MITRE ATT&CK [11] framework. This threat overlay capability provides contextual insight into how software assets may be targeted or misused within operational environments and supports more informed risk assessment.

Asset Threat Overlay is a capability that could also be extended to a centralized government-wide firmware and hardware acquisition service.

2.1.2.4. Asset Provisional Assessment

Building on standardized security representations and integrated vulnerability and threat data, BloSS@M supports automated and AI-assisted *provisional security assessments*. A *provisional security assessment* in this context refers to a preliminary risk determination for a component

of a system that has not yet been completed. It is based on initial security controls testing, risk analysis, and OSCAL [7] Component Definitions documentation. Full assessment and authorization activities take place later in the context of the system that the asset will be supporting.

These assessments leverage agentic and generative AI techniques to analyze implemented controls documented in a digital format (e.g., OSCAL [7] Component Definitions) and document known threats and respective mitigations that leverage MITRE ATT&CK [11] and D3FEND [12]. The resulting assessment outputs are intended to support continuous monitoring, comparative risk evaluation, and decision-making while complementing formal authorization and risk management processes.

2.2. Blossom Members ATO

Since BloSS@M enables the processing, storage, and sharing of federal information across organizational boundaries, all government agencies that use a BloSS@M service remain fully responsible for deploying, securing, and managing their members. They are also subject to the Federal Information Security Modernization Act (FISMA) [13] and must assess and authorize to operate (ATO) [14] their member and all other members that join the BloSS@M network. The use of distributed ledger technology does not diminish agency accountability for protecting information systems and information, nor does it transfer authorization responsibility to the service operator or to other participating members.

Given the federated and dynamic nature of BloSS@M, manually performing and maintaining an ATO [14] for each new member system is not scalable, timely, nor sustainable and introduces unacceptable risks to mission operations. Therefore, automated security assessments, continuous monitoring, and machine-readable authorization artifacts are mandatory to support near-real-time risk determination and authorization decisions. The adoption of OSCAL is recommended to enable standardized, interoperable, and automated exchanges of system security plans, control assessments, and authorization data, thereby supporting consistent FISMA compliance while preserving the agility required for a shared, multi-agency blockchain service. IR 8500B elaborates on the continuous ATO (cATO)[15] and provides an example of how to achieve it.

3. Leveraged Technologies

The following subsections describe the integrated technologies that enable BloSS@M's capabilities.

3.1. Blockchain and Next Generation Access Control

BloSS@M relies on blockchain technology as a foundational mechanism for maintaining a shared, verifiable record of software asset life cycle information across organizational boundaries. A blockchain is a distributed digital ledger that records transactions in a manner that is resistant to unauthorized modification. Transactions are grouped into blocks and cryptographically linked to preceding blocks, forming an append-only record that enables the detection of unauthorized changes.

Transactions recorded on a blockchain are digitally signed using public and private cryptographic keys to provide authenticity, integrity, and non-repudiation assurances. Once recorded and validated by the network, transactions cannot be altered or removed without detection. These properties enable participating entities to rely on the integrity of recorded information without requiring full mutual trust or a single centralized authority.

Blockchain systems may incorporate encryption mechanisms to protect data confidentiality during storage and transmission. In addition, access to blockchain-recorded information can be governed through policy-based access control mechanisms that determine which participants are permitted to view, submit, or interact with specific data elements. Such controls are particularly relevant in IT/OT and federal environments, where multiple stakeholders require differing levels of visibility, responsibility, and authority over asset-related information.

Blockchain networks typically operate across multiple participating nodes that each maintain a replicated copy of the ledger. This distributed architecture enhances availability, resilience, and fault tolerance by reducing dependence on any single system or organization. Consensus mechanisms are used to ensure that participating nodes agree on the validity and ordering of transactions before they are recorded to the ledger.

Blockchain systems commonly exhibit the following attributes:

- **Ledger:** A persistent, append-only, and tamper-resistant record of transactions.
- **Security:** Cryptographic mechanisms protect data integrity, support authentication, and enable non-repudiation.
- **Shared:** A ledger is accessible to authorized participants within the network according to defined governance and access policies.
- **Distributed:** Replicated copies of the ledger are maintained across multiple nodes to improve availability and resilience.

To enable data interactions, blockchain implementations use smart contracts, which implement the business logic for a given use case. Smart contracts are packaged into chaincode, which is deployed on the network and executed for a given use case. Any updates to the chaincode

require approval from a configurable number of member organizations that own and maintain peer nodes in order to prevent the deployment of malicious code.

In the context of BloSS@M, these characteristics support the creation of a verifiable and auditable record of software asset information, including license allocation and life cycle events, while enabling controlled information sharing among government entities and other authorized participants.

3.1.1. Permissioned Blockchain

There are two primary types of blockchain architectures: *permissionless* and *permissioned*. In a *permissionless blockchain*, network participation is open to any entity without prior authorization. Participants can anonymously read data from or write transactions to the ledger, relying on consensus mechanisms such as proof of work (PoW) to establish trust. While this model supports decentralization and transparency, it often results in limited scalability, high computational costs, and reduced data confidentiality.

In contrast, a *permissioned blockchain* restricts participation to authenticated entities. Only authorized users or organizations can read from or write to the ledger. This restricted-access model enables enhanced control over data visibility and transaction validation and supports improved confidentiality, integrity, and operational efficiency.

Permissioned managed blockchains (PMBs) extend these concepts by incorporating mechanisms for governance, user management, and system monitoring. Compared with permissionless blockchains, PMBs typically offer:

- **Higher scalability and performance** — Achieved by utilizing cryptographic proof of authorization rather than computationally intensive consensus algorithms, such as PoW
- **Reduced transaction costs** — Lower resource consumption and streamlined network operations
- **Granular access control** — Fine-grained permissions applied at the user, organization, or channel level
- **Configurable consensus mechanisms** — Enable adaptation to specific operational or compliance requirements
- **Simplified governance and onboarding processes** — Clearly defined participant roles and controlled membership management
- **Resilience against consensus manipulation** — Membership and transaction validation governed by authorized entities rather than arbitrary network participants
- **Role-based management interfaces** — Facilitated by system oversight and maintenance

The capabilities offered by BloSS@M require strong assurance of data confidentiality and integrity across multiple organizations. To meet these requirements, BloSS@M uses a *permissioned blockchain* architecture that is capable of supporting *multiple isolated ledgers* within the same network. This can be accomplished using *channel-based segmentation*, where

each channel maintains a distinct ledger with restricted access to a defined subset of authorized participants. This design ensures that sensitive data is disclosed only to relevant entities while maintaining a verifiable record of transactions and preserving system integrity.

3.1.2. Next Generation Access Control

To enhance access control for data that is generated and maintained by BloSS@M, NIST's Next Generation Access Control (NGAC) [15] is implemented into the smart contracts of the blockchain's chaincode. NGAC is an American National Standards Institute (ANSI)[1] attribute-based access control (ABAC) standard that uses a graph structure to represent complex access control policies and relationships between users, objects, and their attributes.

NGAC also defines a standard functional architecture with the following components:

- Policy Enforcement Point (PEP) [16]
- Policy Decision Point (PDP) [16]
- Policy Administration Point (PAP) [16]
- Event Processing Point (EPP) [16][17]
- Policy Information Point (PIP) [16]
- Resource Access Point (RAP) [16]

Each component has a distinct role in creating a flexible and scalable system that can handle a wide range of access control policies.

There are five types of policy elements represented as nodes in the graph:

1. User (human or non-person entities)
2. Object (logical representations for physical resources, regardless of format or location)
3. User Attribute
4. Object Attribute
5. Policy Class

Relationships between these elements include:

- **Assignment:** Builds a hierarchy by linking elements and grouping users and objects under attributes. Attributes can nest within other attributes or policy classes that define high-level policy domains (e.g., role-based access control [RBAC], discretionary access control [DAC]).
- **Association:** Represents a connection between a user attribute and another user or object attribute, shown as an edge in the policy graph. Associations grant access rights

[1] See https://www.ansi.org.

from users in the source attribute to policy elements (e.g., resources) in the target attribute.

- **Prohibition:** Explicitly denies access rights and is checked after associations for finer control.
- **Obligation:** Specifies event patterns and the administrative actions that are triggered when those events occur, managed by the Event Processing Point (EPP) [17].

3.2. Open Security Controls Assessment Language

OSCAL [7] is a set of standardized, machine-readable models that represent cybersecurity and privacy information across the system life cycle. OSCAL enables the structured expression of security controls, control implementations, assessment results, and authorization artifacts using common formats, such as XML, JSON, and YAML. By providing consistent data representation, OSCAL supports automation, interoperability, and the reuse of security information across tools, organizations, and governance processes.

OSCAL defines a suite of eight machine-readable models that collectively support the end-to-end automation of security control selection, implementation, assessment, and authorization:

1. **Catalog Model** — Provides a structured representation of security controls and control enhancements
2. **Profile Model** — Supports the tailoring and composition of control baselines
3. **Component Definition Model** — Describes reusable system components and the controls they implement
4. **System Security Plan (SSP) Model** — Documents system-specific control implementations
5. **Assessment Plan Model** — Defines the scope and methods for evaluating control effectiveness
6. **Assessment Results Model** — Captures the outcomes of security assessments
7. **Plan of Action and Milestones (POA&M) Model** — Tracks identified risks and remediation activities
8. **Control Mapping Models** — Allows for documenting the alignment between requirements and controls from different regulatory frameworks or versions of the same framework with mathematical precision

Together, these models enable standardized data exchange, continuous monitoring, and automated authorization workflows in support of the NIST Risk Management Framework (RMF) and FISMA compliance.

Within the BloSS@M solution, OSCAL serves as a foundational mechanism for representing and exchanging authoritative security information associated with software assets and supporting infrastructure. Similar types of information can be generated for hardware or firmware assets

for any centralized assets acquisition and management service. OSCAL also supports asset-level security configuration visibility, as documented by vendors of respective software assets, and enables security assessment automation and cATO aligned with ongoing risk management practices. Each BloSS@M participant's infrastructure must demonstrate cATO to meet FISMA and BloSS@M MOU [18] requirements (see Sec. 4.2 for details).

3.3. SWID for Assets Traceability

SWID tagging [19] was developed to improve software discovery, identification, and contextualization as well as the accuracy and reliability of software asset management processes. SWID tags provide the following benefits to software creators, publishers, buyers, and users:

- Indicate the presence of software on a device to aid discovery
- Provide a product label to support identification
- Include product details (e.g., name, version) to enable contextualization

When software is installed on a device, a unique ISO SWID tag is generated and serves as a digital fingerprint for the software. This tag is created and stored on the device, and a corresponding entry is made on the BloSS@M blockchain as a permanent and tamper-proof record of the software installation. As part of the leasing process, the device is required to report the generated SWID tag back to the blockchain, thus ensuring that the software is properly accounted for and tied to the relevant license. When a license is released by the end user, the software must be uninstalled, and the corresponding SWID tag is removed from the device's registry. This removal is then reflected on the blockchain by updating the record to indicate that the software is no longer in use. By leveraging SWID tagging, BloSS@M ensures that software usage is accurately tracked and managed to prevent unauthorized use and ensure compliance with licensing agreements.

SWID tags support stakeholders throughout the system development life cycle (SDLC) with:

- Continuous and accurate identification of software to be managed
- Reliable tracking of software changes on devices using consistent identifiers
- Standardized information exchange between producers and consumers
- Automated compliance tracking for software licenses and usage
- Collection of detailed asset data, including what software is deployed and where
- Identification of all parties involved in installing, licensing, maintaining, and managing software

There are four types of SWID tags: *corpus, primary, patch,* and *supplemental*. Corpus, primary, and patch tags indicate the presence of software, while supplemental tags provide additional information that is not included in the other three tags.

1. **Corpus SWID tags** are used by vendors and distributors to describe and identify software products before installation. They help verify the software product and authenticate the organization releasing it. These tags can also be used by installation tools to confirm the integrity of the software and validate that a license exists prior to installation.
2. **Primary SWID tags** identify and describe software after it has been installed. They typically include the product name, a unique tag identifier, and information identifying the tag's creator. The tag identifier can serve as a proxy for the software product, allowing inventory tools to identify and report software using the tag.
3. **Patch SWID tags** describe individual patches. When a patch is installed, its corresponding SWID tag is also added to the device. These tags contain data about the patch itself, not the software that it updates.
4. **Supplemental SWID tags** add information after installation. They can be created by users or software management tools and may include details like license keys or contact information [21]. These tags can be installed at any time but can only be edited by their creator.

While some vendors provide tools to manage licenses, updates, patches, and configurations for their own products, organizations often use a wide variety of such tools across different vendors. This fragmented approach increases the risk of human error and resource limitations, making it harder to manage software effectively. As a result, patches may be delayed, configurations may drift, and software licenses may be underused or duplicated. To address this challenge, a unified mechanism is needed to provide a comprehensive view of software across the enterprise, regardless of vendor. One solution is the use of ISO/IEC 19770-2:2015 [20], a standard published by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to define structured metadata for software identification (SWID) tags [19]. These tags provide a consistent format for describing software products to enable better management, tracking, and auditing across diverse software environments.

A SWID tag document contains a structured set of data elements that identify the software, describe its version, list the organizations and individuals involved in its production and distribution, detail the artifacts that make up the product, describe relationships between software products, and include other relevant metadata. This information supports software asset management and security tools by enabling automation across the software development life cycle. SWID tags help automate:

- Software inventory in software asset management
- Vulnerability assessments of installed software
- Detection of missing patches

- Configuration checklist assessments
- Software integrity verification
- Creation of execution and installation allowlists and denylists
- Other operational and cybersecurity functions

NIST recommends that software producers and standards bodies adopt ISO/IEC 19770-2:2015, including organizations such as the Trusted Computing Group (TCG) [21] and the Internet Engineering Task Force (IETF) [22]. Building on this standard, NIST, the Department of Homeland Security (DHS), and the National Security Agency (NSA) jointly published NIST Internal Report (NIST IR) 8060, *Guidelines for the Creation of Interoperable SWID Tags* [23]. This report provides detailed guidance for creating SWID tags that support cybersecurity applications. To assist in implementing these guidelines, NIST also released a SWID Tag validation tool, which enables tag producers to verify compliance with both ISO/IEC 19770-2:2015 and IR 8060. Additionally, NIST collaborated with TagVault.org, to produce SWID Tag signing guidelines [23], which describe the use of XML digital signatures [25] to verify the source and integrity of SWID tags. These efforts collectively enhance trust, interoperability, and the consistent management of software assets across diverse environments.

3.4. Asset Security

3.4.1. Secure Configuration and Controls Satisfaction With OSCAL

The *asset security configuration and control implementation* capability provided by BloSS@M enables software vendors and other asset providers to supply standardized, machine-readable security information that documents the controls and safeguards implemented by a given software asset in accordance with applicable policies and regulations. This capability supports the exchange of security-related information using the OSCAL [7] Component Definition model [9], which is designed to describe the security characteristics, implemented controls, and secure configuration guidance associated with reusable system elements, including software products and services.

In addition to documenting implemented controls, OSCAL Component Definitions support the representation of secure configuration options and guidance that describe how an asset should be deployed and configured to achieve specific security or privacy objectives. This enables organizations to consistently interpret vendor-provided security information and to integrate that information into automated security assessment and authorization workflows.

OSCAL Component Definition artifacts may describe:

- Information about the organization that develops, provides, and maintains the software asset
- Characteristics of the software asset, including name, version, model, and last-modified date

- Security and privacy controls or control statements that are implemented or supported by the asset
- Configuration settings and implementation guidance for achieving specific security or privacy objectives
- Assessment methods, test cases, or scripts that may be used to validate control implementation.

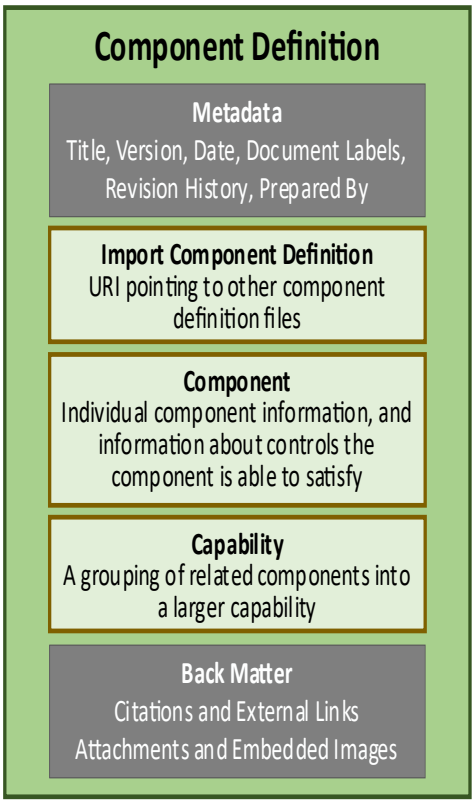


Fig. 3. OSCAL Component Definition model

As shown in Fig. 3, an OSCAL Component Definition model is organized as follows:

- **Metadata:** Metadata syntax is identical and required in all OSCAL models. It includes information such as the document's title, publication version, publication date, and OSCAL version. Metadata is also used to define roles, parties (i.e., people, teams and organizations), and locations.
- **Import:** Identifies a component definition from another resource from which related information is referenced.
- **Component:** Provides information about a defined component that can be part of an implemented system.
- **Capability:** Identifies a capability as a grouping of multiple components or capabilities.
- **Back Matter:** Back matter syntax is identical in all OSCAL models. It is used for attachments, citations, and embedded content, such as graphics.

The OSCAL Component Definition model also supports grouping related components into *capabilities*, which allows asset providers to document how multiple components can be used together to satisfy security or privacy controls that may not be fully implemented by a single component alone. This capability-based composition supports more accurate representation of complex software products and services.

Component Definitions further support the composition of a *System Security Plan (SSP)* [26] by enabling the reuse of component-level control implementation statements across systems. When developing an OSCAL SSP, organizations can reference and incorporate Component Definition artifacts to describe how system elements — including software assets that are managed through BloSS@M — contribute to the overall system control implementation. This approach reduces duplication, improves consistency, and automatically updates system-level documentation to reflect component security information.

Within a Component Definition, individual components may be used to document and share:

- Evidence of compliance with specific security requirements for hardware or software, such as the use of FIPS 140-2-validated cryptographic modules
- Information that describes how a hardware, software, or service implements or supports specific security or privacy controls
- Descriptions of policies or procedures that contribute to the implementation of security or privacy controls

By leveraging OSCAL Component Definitions within BloSS@M, organizations can improve the quality, reuse, and automation of security documentation while supporting scalable system authorization and continuous risk management practices.

3.4.2. NVD and Assets Vulnerabilities

The ability to identify, correlate, and assess publicly disclosed software vulnerabilities is a critical component of effective software asset management and supply chain risk management. Within the BloSS@M service, the *asset vulnerability identification* capability standardizes vulnerability information and automated access to authoritative data sources in support of continuous awareness of software-related security risks.

The following sections describe how BloSS@M leverages the Common Vulnerabilities and Exposures (CVE) program [27] and the NVD [10], including automated data retrieval mechanisms, to enable scalable and timely vulnerability analysis across managed software assets.

3.4.2.1. Leveraging CVEs and the NVD Process

BloSS@M leverages the CVE program [27] and the NVD [10] to support the consistent identification and analysis of publicly disclosed software vulnerabilities. The CVE program provides a standardized catalog of vulnerability identifiers that are associated with specific software, firmware, and hardware products. Each vulnerability is assigned a unique CVE

identifier to enable consistent referencing and information exchange across vendors, tools, and organizations. The CVE program is maintained by the MITRE Corporation and sponsored by DHS and the Cybersecurity and Infrastructure Security Agency (CISA).

The NVD builds upon CVE data by enriching vulnerability records with additional information, including affected products, severity metrics, impact data, and reference links. The Common Platform Enumeration (CPE) program [28] complements the CVE by providing standardized naming conventions for IT products and platforms. Together, the CVE, NVD, and CPE form key components of the Security Content Automation Protocol (SCAP) [29], which supports the automation of vulnerability management and assessment activities in federal environments.

Within BloSS@M, CVE, NVD, and CPE data are used to associate managed software assets with known vulnerabilities in a standardized and machine-readable manner. This association supports automated vulnerability awareness and risk analysis as part of software asset life cycle management.

3.4.2.2. Automated Vulnerability Queries and NVD Integration

The NVD provides publicly available *application programming interfaces (APIs)*[2] that enable programmatic access to vulnerability data. These interfaces support structured queries based on various parameters, such as the CVE identifier, CPE name, vendor, product, and software version. BloSS@M integrates with NVD APIs to support the automated and on-demand retrieval of vulnerability information that is relevant to software assets managed within the platform.

The NVD API integration within BloSS@M follows a structured data flow:

- **Asset Identification**
  Leveraging OSCAL component definition artifacts, software assets registered in BloSS@M are associated with identifying metadata, including product name, version, and standardized CPE identifiers where available.
- **Query Construction**
  Based on asset metadata, BloSS@M constructs NVD API queries using appropriate search parameters (e.g., CPE name, CVE identifier) to retrieve vulnerability records that are relevant to the asset.
- **API Invocation**
  BloSS@M submits requests to NVD APIs using supported query mechanisms. These requests are executed on demand or as part of scheduled monitoring activities.
- **Data Retrieval and Normalization**
  NVD API responses are returned in structured formats and include CVE identifiers, affected versions, severity metrics (e.g., CVSS scores), and reference information. BloSS@M normalizes this data to ensure consistent representation across assets and vendors.

[2] See https://nvd.nist.gov/developers/vulnerabilities?ref=packetcoders.io.

- **Correlation and Association**
  Retrieved vulnerability data is correlated with the corresponding software assets within BloSS@M to enable traceability between assets, vulnerabilities, and security information.
- **Downstream Analysis and Use**
  Associated vulnerability data is made available for comparative analysis, risk evaluation, and integration with other BloSS@M capabilities, including threat mapping, AI-assisted provisional assessments, and continuous authorization activities.

By integrating automated NVD API queries into its technical architecture, BloSS@M supports timely vulnerability awareness, reduces manual data collection, and enables scalable vulnerability monitoring that is aligned with continuous risk management and supply chain security objectives.

3.4.3. Leveraging MITRE ATT&CK and D3FEND for Software Asset Resilience Assessment

BloSS@M leverages the MITRE ATT&CK [11] and MITRE D3FEND [12] knowledge bases to support structured assessments of how resilient a software asset is to known and documented adversary behaviors. These frameworks provide complementary perspectives: MITRE ATT&CK catalogs observe adversary tactics, techniques, and procedures (TTPs), while MITRE D3FEND describes defensive techniques and countermeasures that can be used to detect, prevent, or mitigate those adversary behaviors.

Within BloSS@M, a software asset resilience assessment begins with the identification of security-relevant characteristics and implemented controls associated with the asset. This information is derived from authoritative sources, including OSCAL Component Definition [9] artifacts that document control implementations and configuration guidance. BloSS@M maps these documented controls to the defensive techniques described in MITRE D3FEND to establish a structured representation of the defensive capabilities provided by the software asset and its expected deployment configuration.

In parallel, BloSS@M uses the MITRE ATT&CK framework to identify relevant adversary techniques that are applicable to the software asset based on its function, deployment context, and known vulnerability profile. This threat context may be further informed by vulnerability data obtained from the NVD, which can indicate which ATT&CK techniques are more likely to be exploited for a given asset.

By correlating documented defensive capabilities (via D3FEND mappings) with applicable adversary techniques (via ATT&CK mappings), BloSS@M can assess the degree to which a software asset provides coverage against known threats. Gaps between applicable ATT&CK techniques and available D3FEND-based mitigations are identified as areas of reduced resilience. Conversely, alignment between implemented controls and relevant defensive techniques indicates stronger resistance to specific classes of adversary behavior.

This correlation supports comparative analysis across software assets that provide similar functionality, enabling BloSS@M to evaluate relative resilience based on documented control

implementations and threat coverage. The resulting analysis can be used to inform software selection, configuration decisions, and risk prioritization as well as support AI-assisted provisional assessments and continuous monitoring activities. By grounding resilience assessment in widely adopted, community-maintained threat and defense frameworks, BloSS@M enables a transparent and repeatable approach to evaluating software asset security in the context of known and evolving threats.

4. Technologies Integration Into Blossom

Table 1 provides traceability between BloSS@M capabilities (see Sec. 2.1) and the primary data sources and standards used to support each capability. This traceability is intended to clarify how authoritative and standardized information is leveraged to enable automation, comparability, and risk-informed decision-making.

Table 1. BloSS@M capabilities-to-data-source mapping

Capability Category	Capability	Primary Data Sources	Purpose
Operational	Asset Discovery	Vendor software catalogs; BloSS@M blockchain ledger	Identify available software assets and maintain authoritative inventory visibility
Operational	Asset License Cost Visibility	Vendor-provided pricing and licensing terms; QSMO procurement records	Support cost comparison, budgeting, and life cycle cost analysis
Operational	Asset Comparison	Aggregated asset metadata; license cost data; OSCAL Component Definitions	Enable the comparative evaluation of software providing similar functionality
Operational	Asset Leasing	Blockchain transaction records; license allocation metadata	Manage the allocation, reassignment, and auditability of software usage rights
Technical	Asset Security Configuration	OSCAL Component Definition artifacts	Represent implemented security controls and configuration characteristics in a machine-readable format
Technical	Asset Vulnerability Identification	NVD	Identify known vulnerabilities that affect software assets and assess exposure
Technical	Asset Threat Overlay	MITRE ATT&CK framework	Map software assets and control implementations to known adversary techniques
Technical	Asset Mitigation Mapping	MITRE D3FEND framework	Associate defensive measures and mitigations with documented controls
Technical	Asset Provisional Assessment (AI-Based)	OSCAL Component Definitions; NVD; MITRE ATT&CK; MITRE D3FEND	Generate automated, provisional assessments of security posture and risk alignment

4.1. Information Access Control in BloSS@M

BloSS@M's permissioned blockchain requires two distinct private channels, each governed by its own set of smart contracts that implement NGAC:

- **Authorization channel:** Dedicated to the assessment and authorization (A&A) process, where peer nodes prove and continuously maintain their security posture in line with the FISMA [13]
- **Assets channel:** Focused on the core business functions of BloSS@M to support the implementation of its operational and technical capabilities

The authorization channel supports automated security assessments, member authorization, and cATO requirements. Meanwhile, the asset channel supports BloSS@M services and its core capabilities to allow users to discover, compare, and securely lease licenses for software assets and to manage such assets during their use time.

BloSS@M relies on the existence of a designated *service provider member* who initiates these channels and utilizes the NGAC [16] policy. This designated member grants necessary privileges in both the authorization and asset channels to all organizational members that join the network and deploy their individual peer nodes.

Building trust among peers through automated authorization and accreditation and continuous authority to operate (C-ATO) [15] processes is essential for securing business operations. Therefore, the a*uthorization channel* governs the activities on the *assets channel* and controls which peer nodes can commit business transactions. IR 8500B will provide more details on this proposed approach.

4.2. Onboarding Process

To use BloSS@M services, users must submit a request to join. This request is used to create an account for the entity, which can be an organization, department, or individual.

The authorization channel acts as a security boundary for the asset channel. Members must prove and continuously maintain a satisfactory security posture or risk losing access to the asset channel smart contracts. When interacting with the *authorization chaincode*, the first step for a member is to read and sign a Memorandum of Understanding (MOU) [3], which defines expected member behavior within the network. After signing, the member is initialized in an unauthorized state until they submit their *ATO attestation* and any supporting documents stipulated in the MOU. This information is securely stored with access strictly controlled by the *authorization channel's* NGAC [16] policy to ensure that only authorized members can query it.

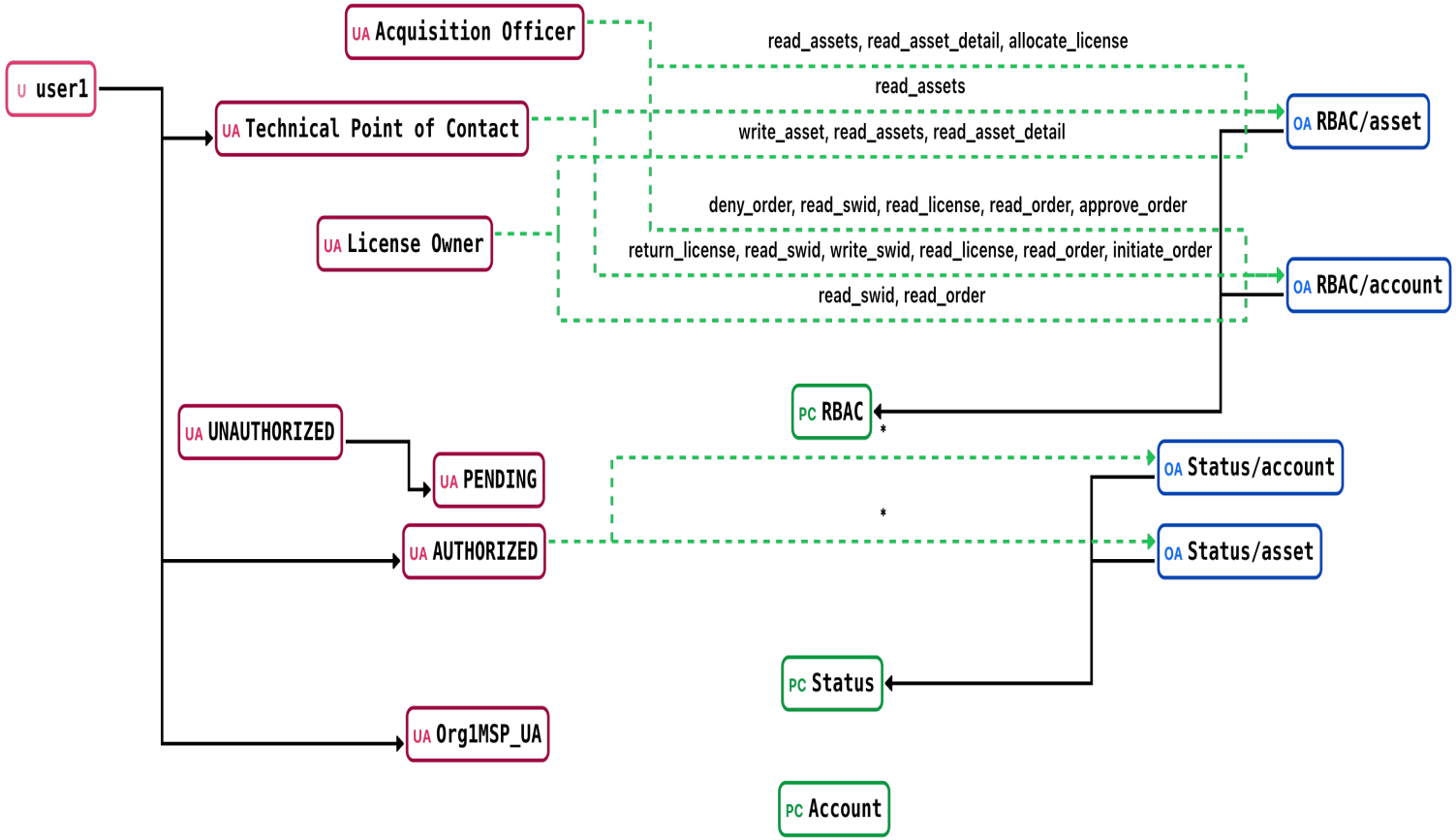
Already authorized members review submitted documents and vote to approve or revoke a member's *authorized status*. Changing a member's status requires a majority vote from authorized members on the channel, including the ability to revoke the service provider's authorization if they fail to maintain adequate security. If a member's authorized status is revoked, they immediately lose access to all smart contracts except the ability to update their new ATO attestation. Updated document is reviewed and, if acceptable, approved by a majority vote, which allows the member to regain full authorization and access to smart contracts across both channels: the *authorization channel* to exercise their right to review and vote on new BloSS@M members and the *assets channel* to use the service.

In the *asset channel*, smart contracts enable the *service provider member* to distribute software asset licenses upon request and the completion of the financial transaction. The smart contracts of the *asset channel* also facilitate the leasing and releasing of these licenses by other network members. Licensed assets obtained by members are securely and confidentially stored in BloSS@M, and the executed smart contract generates an immutable, confidential *ledger*. The NGAC policy enforces strict isolation to prevent members from accessing one another's leased license information.

After obtaining licenses, members report their installation using SWID tags, which remain accessible only to the service provider. To support enhanced traceability, BloSS@M stores a hash of each license's information as a *ledger* to preserve the history of leasing and associated SWID tags. This history can only be tracked by the *service provider member*, as the license hash is salted with a secret known solely to them to ensure secure and private tracking.

One key benefit of using a graph-based access control models (e.g., NGAC) is easy policy review. Traversing the graph enables a comprehensive understanding of the current access permissions at any time. This also supports features like Just-in-Time (JIT) evaluations, where users who are not stored in the policy can still be authorized based on their credentials mapped to the attributes graph. In BloSS@M, JIT is used to effectively separate user management from authorization decisions.

Figure 4 represents the base *asset chaincode* NGAC policy. When this policy is generated in memory, a user is assigned to the attributes that correspond with their provided credentials. In Fig. 4, *user1* is a Technical Point of Contact (TPoC) for the *Org1MSP*, which has a status of *AUTHORIZED*. The target of the decision is determined by the intended action of the user.



762

763 **Fig. 4. Example of asset chaincode NGAC policy**

4.2.1. OSCAL Support for cATO

The onboarding MOU requires all BloSS@M members to maintain predefined security postures. It also requires a set of security controls that must be implemented and assessed. The assessment must be done automatically using OSCAL-formatted documentation for centralized reviewing and ad-hoc auditing. A predefined structured attestation of the completed ATO with supporting documentation must be presented and securely stored by the *authorization channel*. The presented ATO attestation and supporting documentation are then subject to authorized members' review and vote.

cATO emphasizes ongoing assessment, monitoring, and risk-informed decision-making rather than periodic, static authorization events. OSCAL enables this approach by providing machine-readable artifacts that can be updated, exchanged, and evaluated as system conditions change.

Within BloSS@M, OSCAL artifacts are expected to be used to support continuous visibility into the security state of authorized members' peer nodes. As vulnerabilities are disclosed, configurations change, or threat conditions evolve, OSCAL-based representations enable automated updates and analyses without requiring complete reauthorization documentation. This supports near-real-time insight into the control implementation status and risk posture and is the main reason for enforcing its use through the signed MOU.

By integrating OSCAL into BloSS@M, the service enforces continuous ATO for members' peer nodes based on predefined control objectives and assessment methods while enabling standardized data exchange, automated assessment inputs, and traceable security evidence. These enforced processes facilitate more timely risk responses, improve alignment with federal zero trust and secure-by-design initiatives, and support scalable, automation-enabled authorization processes across diverse IT/OT environments.

5. Use-Case-Driven Walkthrough: End-to-End Application of BloSS@M

This section illustrates how a federal agency might apply BloSS@M capabilities across the software asset life cycle, from identification through risk-informed decision-making.

5.1. Use Case Overview

A federal agency requires a commercial off-the-shelf (COTS) software solution to support a mission function (e.g., data analytics). Multiple software products with similar functionalities are available across the federal enterprise, each with different licensing models, costs, and security characteristics. The agency seeks a solution that meets mission needs while minimizing costs and cybersecurity risks.

5.2. Asset Discovery and Initial Filtering

Using BloSS@M, the agency queries the available software asset inventory to identify products that meet predefined functional and policy criteria. BloSS@M returns a list of candidate software assets that are currently available through the enterprise leasing model, including products already licensed for use by other agencies.

5.3. Cost and Leasing Evaluation

For each candidate asset, BloSS@M presents license leasing terms and associated cost information. The agency compares projected usage costs and evaluates whether existing leased licenses can be reassigned or expanded, reducing the need for new procurement actions.

5.4. Security Configuration and Control Review

The agency reviews the security posture of each candidate software asset using OSCAL Component Definition artifacts. These artifacts provide structured descriptions of implemented security controls to enable side-by-side comparisons of declared security capabilities and alignment with organizational control requirements.

5.5. Vulnerability and Threat Analysis

BloSS@M automatically associates each candidate software asset with current vulnerability information from the NVD. In parallel, BloSS@M overlays threat context by mapping documented control implementations to relevant MITRE ATT&CK techniques and corresponding MITRE D3FEND mitigations. This analysis highlights differences in the exposure and defensive coverage of candidate solutions.

5.6. AI-Assisted Provisional Assessment

Based on the aggregated security configuration, vulnerability, and threat data, BloSS@M generates an AI-assisted provisional assessment for each candidate asset. These assessments summarize potential risks, identify control gaps, and evaluate how effectively implemented controls align with known adversary techniques and defensive strategies. The results are provided as decision-support inputs rather than formal authorization artifacts.

5.7. Selection and Ongoing Monitoring

Using the comparative cost, functionality, and risk information provided by BloSS@M, the agency selects the software asset that best meets its mission, budgetary, and cybersecurity requirements. Once deployed, BloSS@M continues to support life cycle management by monitoring license usage, tracking newly disclosed vulnerabilities, and updating threat and mitigation mappings as external data sources evolve.

To summarize, BloSS@M uses the SWID tags (which provide standardized, machine-readable metadata that uniquely identify software products, versions, editions, and patch levels as well as relationships, such as installation, update, and removal events) to continuously monitor license entitlement and consumption. Organizational members could track installation and uninstallation activities across environments and verify the application of updates and security patches throughout the asset lease life cycle to demonstrate that software assets were uninstalled before the licenses are returned to the BloSS@M pool of licenses. By correlating SWID tag data with licensing terms, configuration baselines, and security posture information, organizations can maintain accurate inventories, reduce license compliance risks, and ensure that deployed software remains current and supportable.

The *asset leasing and traceability* capability provided by BloSS@M supports auditability, operational governance, and continuous authorization processes by providing organizational members with near-real-time evidence of software state changes and life cycle events in a standardized and automatable manner and by allowing BloSS@M *service provider members* to maintain accurate inventories of licenses for all available software assets.

6. BloSS@M Benefits

BloSS@M provides secure, robust, and cost-saving software asset license leasing and secure life cycle management for the effective governance of modern IT and OT environments, including the acquisition, deployment, operation, maintenance, and disposition of software assets. As software supply chains become increasingly complex and distributed, organizations require mechanisms that provide authoritative visibility, integrity, and accountability across organizational and contractual boundaries. Distributed ledger technologies offer potential advantages for shared asset tracking and auditability. However, conventional blockchain implementations present limitations when applied to software asset management, particularly with respect to confidentiality, access control, and the privacy of operational data.

BloSS@M addresses these limitations by applying blockchain technology in a manner that is tailored to federal software asset management and security requirements. BloSS@M enables the secure, policy-governed management of software assets and the downstream distribution of leased software licenses across departments, agencies, vendors, and publishers operating in low-trust or trustless environments. By combining tamper-evident ledger capabilities with policy-based access controls, BloSS@M preserves data integrity and availability while supporting confidentiality and least-privilege access to sensitive asset information.

The platform further strengthens trust and assurance through automation. BloSS@M integrates standardized software identification mechanisms (e.g., ISO/IEC 19770-2 SWID tags) to provide continuous, machine-readable evidence of software installation, usage, updates, and removal. These capabilities enhance auditability and accountability so that organizations can detect unauthorized changes, license misuse, or deviations from approved configurations.

In parallel, BloSS@M supports automated security assessments and cATO workflows by integrating vulnerability, threat, and control implementation data from authoritative sources. This reduces manual effort and enables continuous inventory updates, ongoing security assessments, and audits to support timely risk-informed decisions.

To summarize, BloSS@M provides a secure, scalable, and interoperable foundation for software asset life cycle governance across complex and distributed federal IT environments. It unifies asset visibility, security posture awareness, compliance evidence, and automated assurance mechanisms so that agencies can govern their software portfolios more effectively, improve operational efficiency, strengthen cybersecurity and supply chain risk management, and maintain compliance in an evolving threat and policy landscape.

References

[1] National Institute of Standards and Technology (2023) Glossary Operational Technology. Available at https://csrc.nist.gov/glossary/term/operational_technology

[2] Office of Management and Budget (OMB) Circular A-130. Available at https://www.whitehouse.gov/wp-content/uploads/legacy_drupal_files/omb/circulars/A130/a130revised.pdf

[3] Office of Management and Budget (OMB) Memorandum M-13-13. Available at https://obamawhitehouse.archives.gov/sites/default/files/omb/memoranda/2013/m-13-13.pdf

[4] Joint Task Force (2018) Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. (National Institute of Standards and Technology, Gaithersburg MD), NIST Special Publication (SP)NIST SP 800-37r2. https://doi.org/10.6028/NIST.SP.800-37r2

[5] Joint Task Force (2020) Security and Privacy Controls for Information Systems and Organizations. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-53r5. https://doi.org/10.6028/NIST.SP.800-53r5

[6] Quality Service Management Offices (QSMO). Available at https://ussm.gsa.gov/qsmo/

[7] Open Security Controls Assessment Language (OSCAL). Available at https://www.nist.gov/OSCAL

[8] Iorga M, Nguyen M (2025) Charting the Course for NIST OSCAL. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Cybersecurity White Paper NIST CSWP 53. https://doi.org/10.6028/NIST.CSWP.53.ipd

[9] National Institute of Standards and Technology (2025) OSCAL Implementation Layer: Component Definition Model. Available at https://pages.nist.gov/OSCAL/learn/concepts/layer/implementation/component-definition/

[10] National Institute of Standards and Technology (2023) National Vulnerability Database (NVD). Available at https://nvd.nist.gov/

[11] MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK) Available at https://attack.mitre.org/

[12] MITRE D3FEND, A Knowledge Graph of Cybersecurity Countermeasures. Available at https://d3fend.mitre.org/

[13] Federal Information Security Modernization Act. Available at https://www.cisa.gov/topics/cyber-threats-and-advisories/federal-information-security-modernization-act

[14] Licata J, McWhite R, Calloway L, Gilbert D, Anderson M, Snyder J, Miller J (2025) Developing Security, Privacy, and Cybersecurity Supply Chain Risk Management Plans for Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-18r2. https://doi.org/10.6028/NIST.SP.800-18r2.ipd

[15] National Institute of Standards and Technology (2023) Glossary Continuous Authority to Operate (C-ATO). Available at https://csrc.nist.gov/glossary/term/continuous_authority_to_operate

[16] InterNational Committee for Information Technology Standards (2020) INCITS 565-2020 – Information technology – Next Generation Access Control. Available at https://webstore.ansi.org/standards/incits/incits5652020

[17] Johnson A, Dempsey K, Ross R, Gupta S, Bailey D (2011) Guide for Security-Focused Configuration Management of Information Systems. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-128. https://doi.org/10.6028/NIST.SP.800-128

[18] National Institute of Standards and Technology (2025) Memorandum of Understanding or Agreement. Available at https://csrc.nist.gov/glossary/term/memorandum_of_understanding_or_agreement

[19] National Institute of Standards and Technology (2025) Software Identification Tags (SWID). Available at https://csrc.nist.gov/projects/Software-Identification-SWID

[20] International Organization for Standardization/International Electrotechnical Commission (2015) I*SO/IEC 19770-2:2015-Information technology-IT asset management Part 2: Software identification tag* (ISO, Geneva, Switzerland). Available at https://www.iso.org/standard/65666.html

[21] Trusted Computing Group (TCG). Available at https://trustedcomputinggroup.org/

[22] Internet Engineering Task Force (IETF). Available at https://www.ietf.org/

[23] Waltermire DA, Cheikes BA, Feldman L, Witte GA (2016) Guidelines for the Creation of Interoperable Software Identification (SWID) Tags. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency Report (IR) NIST IR 8060. https://doi.org/10.6028/NIST.IR.8060

[24] SWID Tag Signing Guidelines. Available at https://tagvault.org/download/swid-tags-signing-guidelines/

[25] XML Digital Signatures. Available at https://www.w3.org/TR/xmldsig-core1/

[26] National Institute of Standards and Technology (2025) OSCAL Implementation Layer: System Security Plan (SSP) Model. Available at https://pages.nist.gov/OSCAL/learn/concepts/layer/implementation/ssp/

[27] Common Vulnerabilities and Exposures (CVE) (2023). Available at https://cve.org/

[28] National Institute of Standards and Technology (2025) Official Common Platform Enumeration (CPE) Dictionary. Available at https://nvd.nist.gov/products/cpe

[29] Security Content Automation Protocol Version 2 (SCAP v2) (National Institute of Standards and Technology, Gaithersburg, MD). Available at https://csrc.nist.gov/projects/security-content-automation-protocol-v2