Software Configuration Management

Unit II

What is Software Configuration?

- The output of a software process is information that may be divided into three broad categories:
  - Computer programs (both source level and executable)
  - Work products that describe the computer programs (targeted at both technical and end users)
  - Data (contained within the program or external to it)
- The items that comprise all the information produced as part of software process are collectively called as software configuration.

Configuration Management

Definition:

The set of activities that have been developed to manage change throughout the software life cycle.

Purpose:

Systematically *control* changes to the configuration and *maintain* the *integrity* and *traceability* of the configuration throughout the system's life cycle.

What is SCM?

- It is an umbrella activity that is applied throughout the software process.
- Because change can occur anytime, SCM activities are developed to:
  - Identify change
  - Control change
  - Ensure that change is being properly implemented
  - Report changed to others who have interest.

Difference between Software Support and SCM

- Support is a set of software engineering activities that occur after the software has been delivered to the customer and put into operation.
- Software Configuration Management is a set of Tracking and Control activities that are initiated when a software engineering project begins.
- SCM is a set of activities that have been developed to manage change throughout the life cycle of computer software.

What is the Origin of these Changes?

- New business or market conditions bring changes in product requirements.
- New customer needs demand modification of functionality delivered by the products or services delivered by the computer based system.
- Reorganization or business growth causes changes in project priorities or s/w engineering team structure.
- Budgetary constraints cause a change in the definition of the system.

Why Change Happens?

- Change is a fact of life in software development.
  - Customers want to modify requirements.
  - Developers want to modify the technical approach.
  - Managers want to modify the project strategy

- Why all these modifications???
  - As time passes while building the software, all the people involved in it come to know more about what they need, which approach will be the best, how to get it done within time constraints, etc.
  - This additional knowledge is the driving force behind most changes in software development.

Software Configuration Items (SCIs)

- **Definition:** Information that is created as part of the software engineering process.
- Examples:
  - Software Project Plan
  - Software Requirements Specification
    - Models, Prototypes, Requirements
  - Design document
    - Protocols, Hierarchy Graphs
  - Source code
    - Modules
  - Test suite
  - Software tools (e.g., compilers)

Baselines

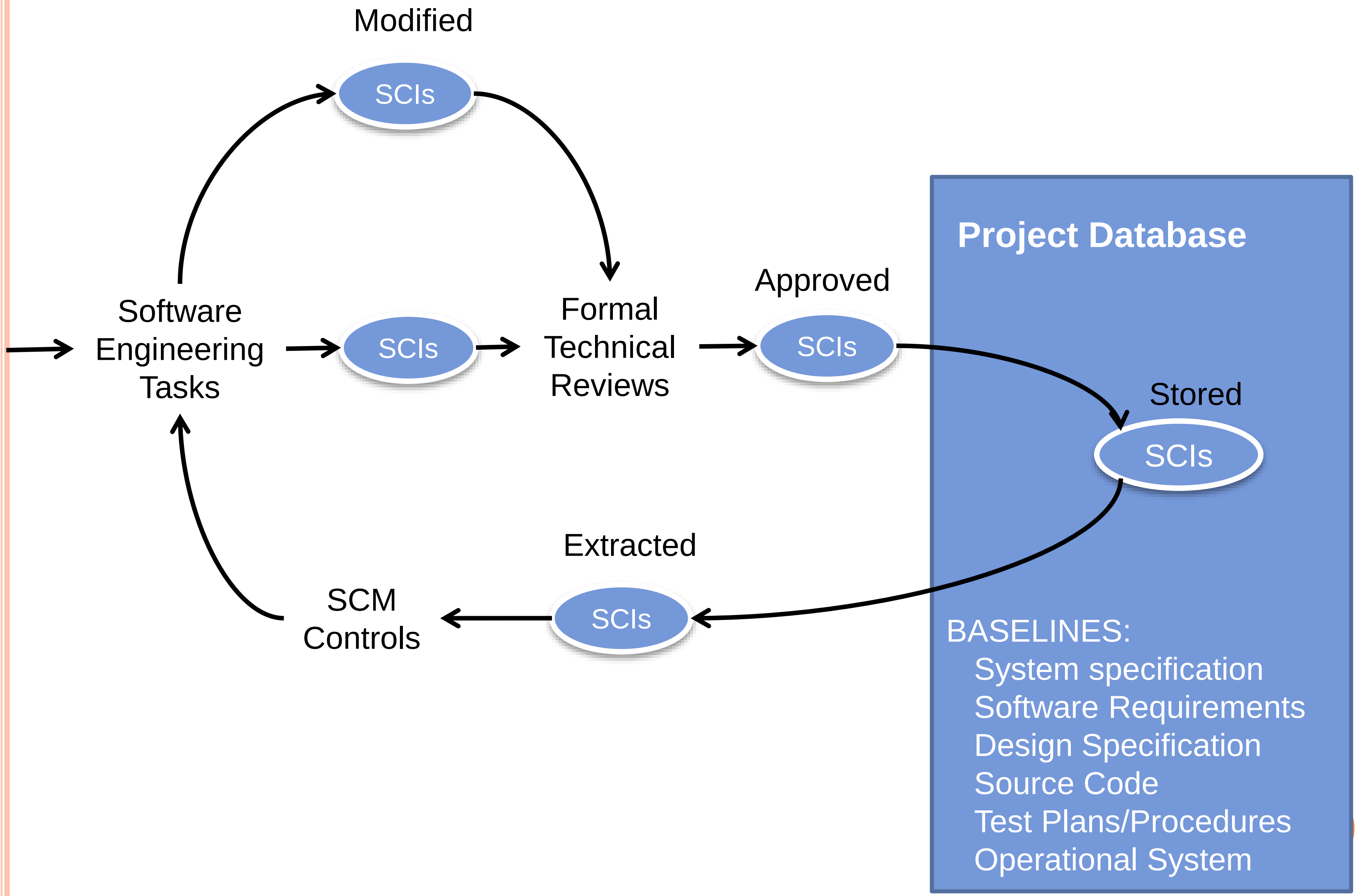
- A Baseline is a software configuration management concept that helps us to control change.
- IEEE Definition: Specification or product that
  - has been formally reviewed and agreed upon,
  - serves as the basis for further development, and
  - can be changed only through formal change control procedures.
- Signals a point of departure from one activity to the start of another activity.
- Helps control change without impeding justifiable change.

Project Baseline

- Central repository of reviewed and approved artifacts that represent a given stable point in overall system development.
- Shared Database for project and kept in consistent state.
- Policies allow the team to achieve consistent state and manage the project.

Baseline Process

1. A series of software engineering tasks produces an SCI
2. The SCI is reviewed and possibly approved.
3. The approved SCI is given a new version number and placed in a project database (i.e., software repository)
4. A copy of the SCI is taken from the project database and examined/modified by a software engineer
5. The baseline of the modified SCI goes back to Step 2



BASELINES:
- System specification
- Software Requirements
- Design Specification
- Source Code
- Test Plans/Procedures
- Operational System

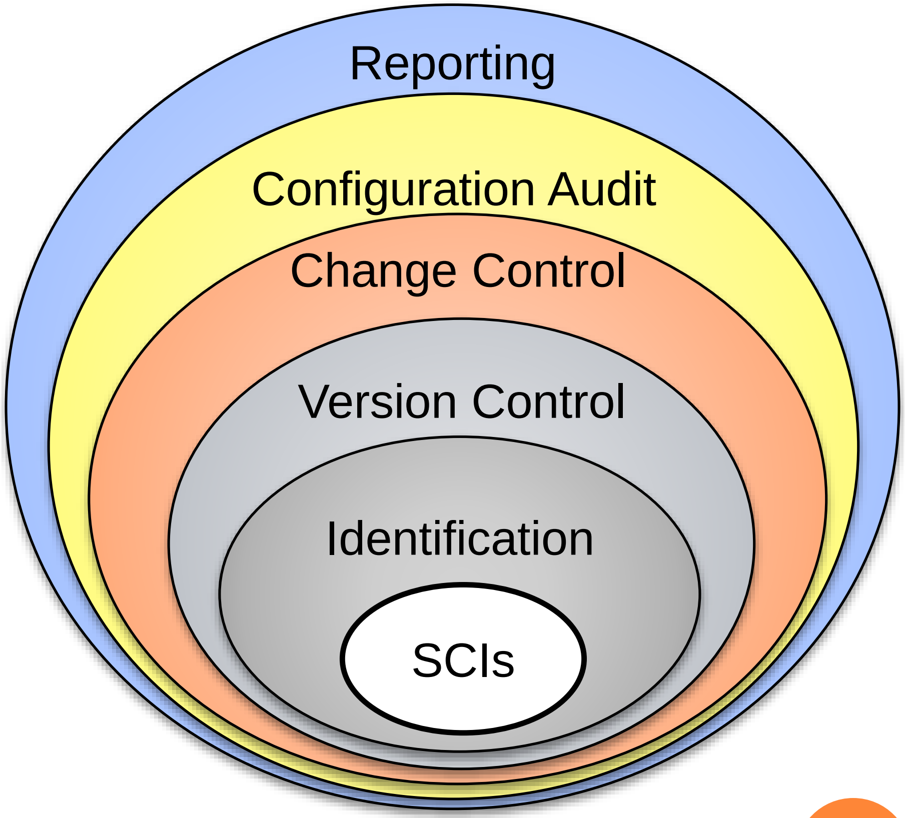
Elements of SCM

There are four elements of SCM:

- Software Configuration Identification
- Software Configuration Control
- Software Configuration Auditing
- Software Configuration Status Reporting

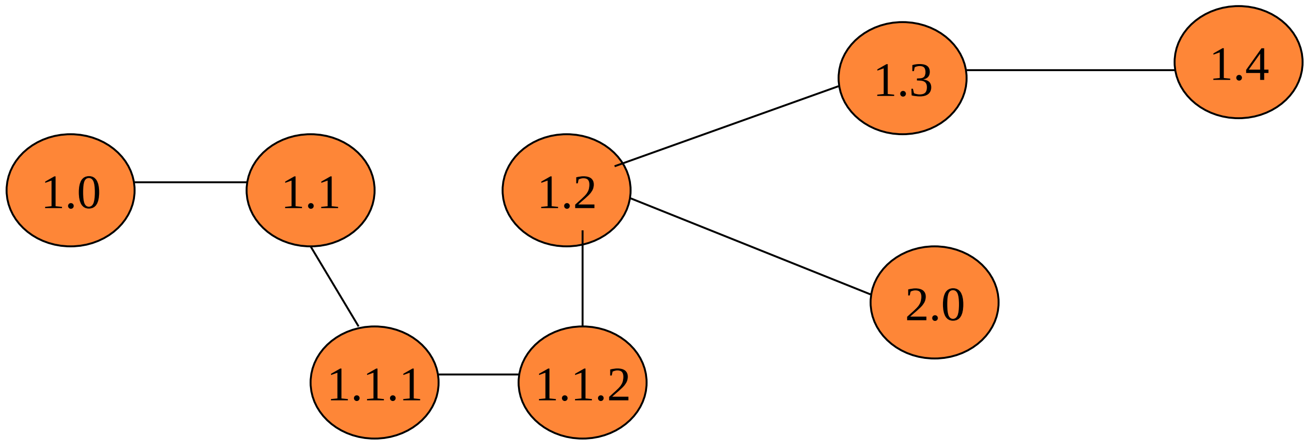
The SCM Process

- The SCM process defines a series of tasks:
  - Identification of objects in the software configuration
  - Version Control
  - Change Control
  - Configuration Audit, and
  - Reporting



The SCM Process: Identification

- Provides labels for the baselines and their updates.
- Evolution graph: depicts versions/variants.



- An object may be represented by variant, versions, and components.

The SCM Process: **Identification**

- Two types of objects can be identified-
  - Basic objects, and
  - Aggregate objects
- A *basic object* is a unit of information created by a software engineer during analysis, design, code, or test.
- For example, a basic object might be a section of requirement specification, part of design model, source code for a component, etc.
- An *aggregate object* is a collection of basic objects and other aggregate objects.

The SCM Process: Identification

- Each object has a set of distinct features that identify it:
  - A **name** that is unambiguous to all other objects
  - A **description** that contains the CSCI type, a project identifier, and change and/or version information
  - **List of resources** needed by the object
  - The **object realization** (i.e., the document, the file, the model, etc.)

The SCM Process: **Software Configuration Control**

Three basic ingredients to SCC

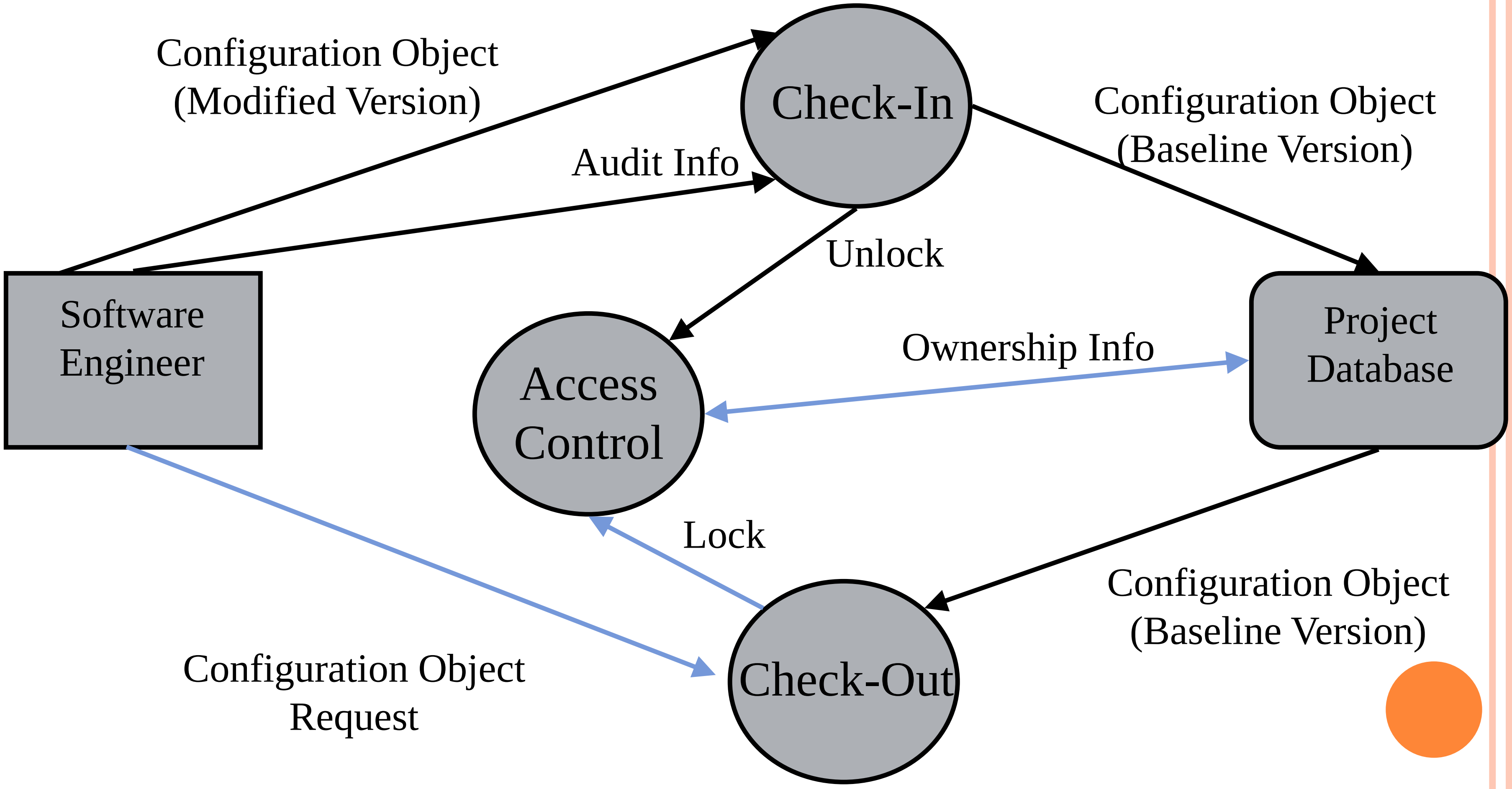
- Documentation for formally precipitating and defining a proposed change to a software system.
- An organizational body (Configuration Control Board) for formally evaluating and approving or disapproving a proposed change to a software system.
- Procedures for controlling changes to a software system.

The SCM Process: **Software Configuration Control**

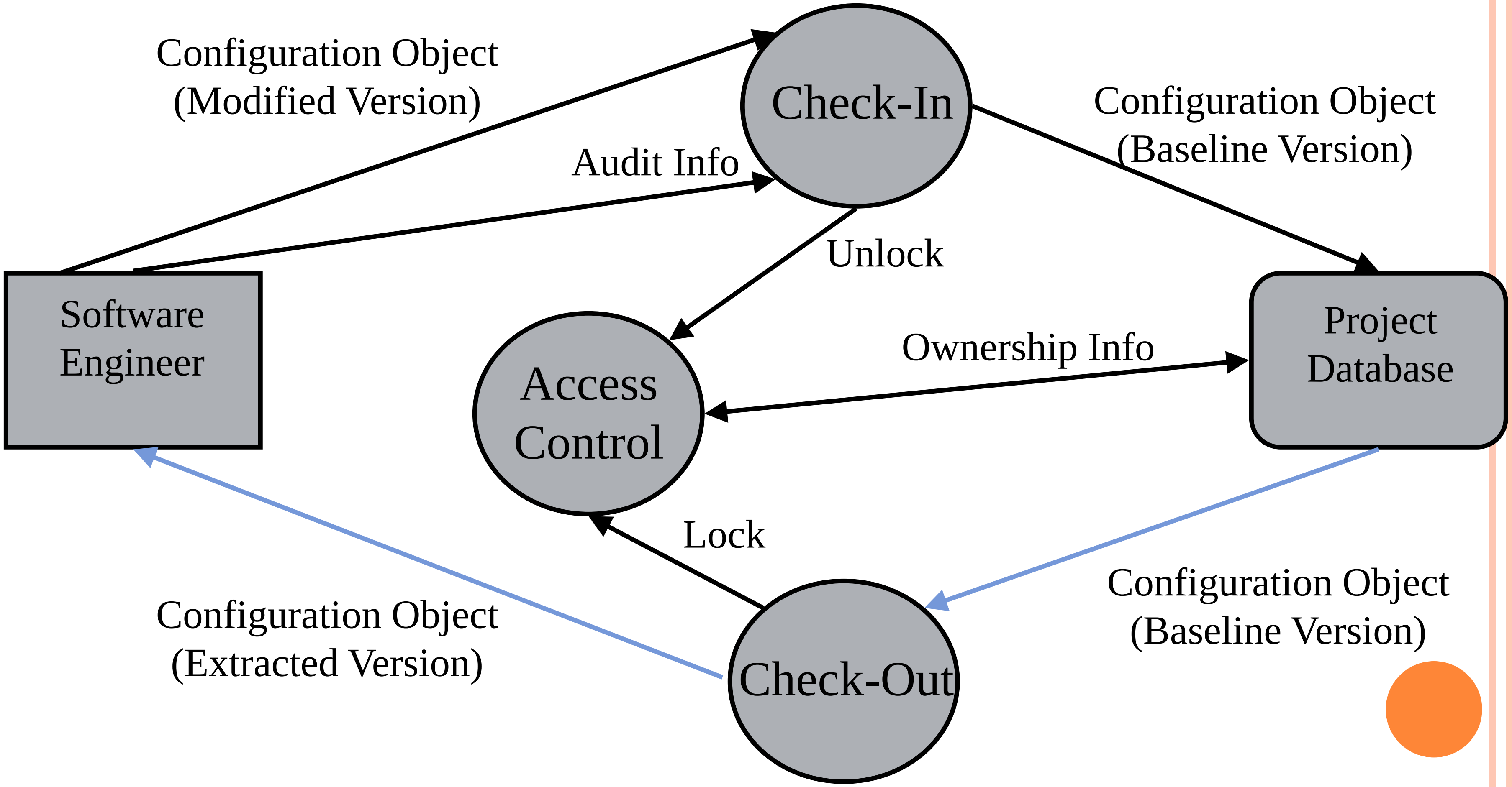
Why is it needed??

- Not all possible changes are beneficial.
- Need a mechanism to control access to different items of the configuration (who can access what).

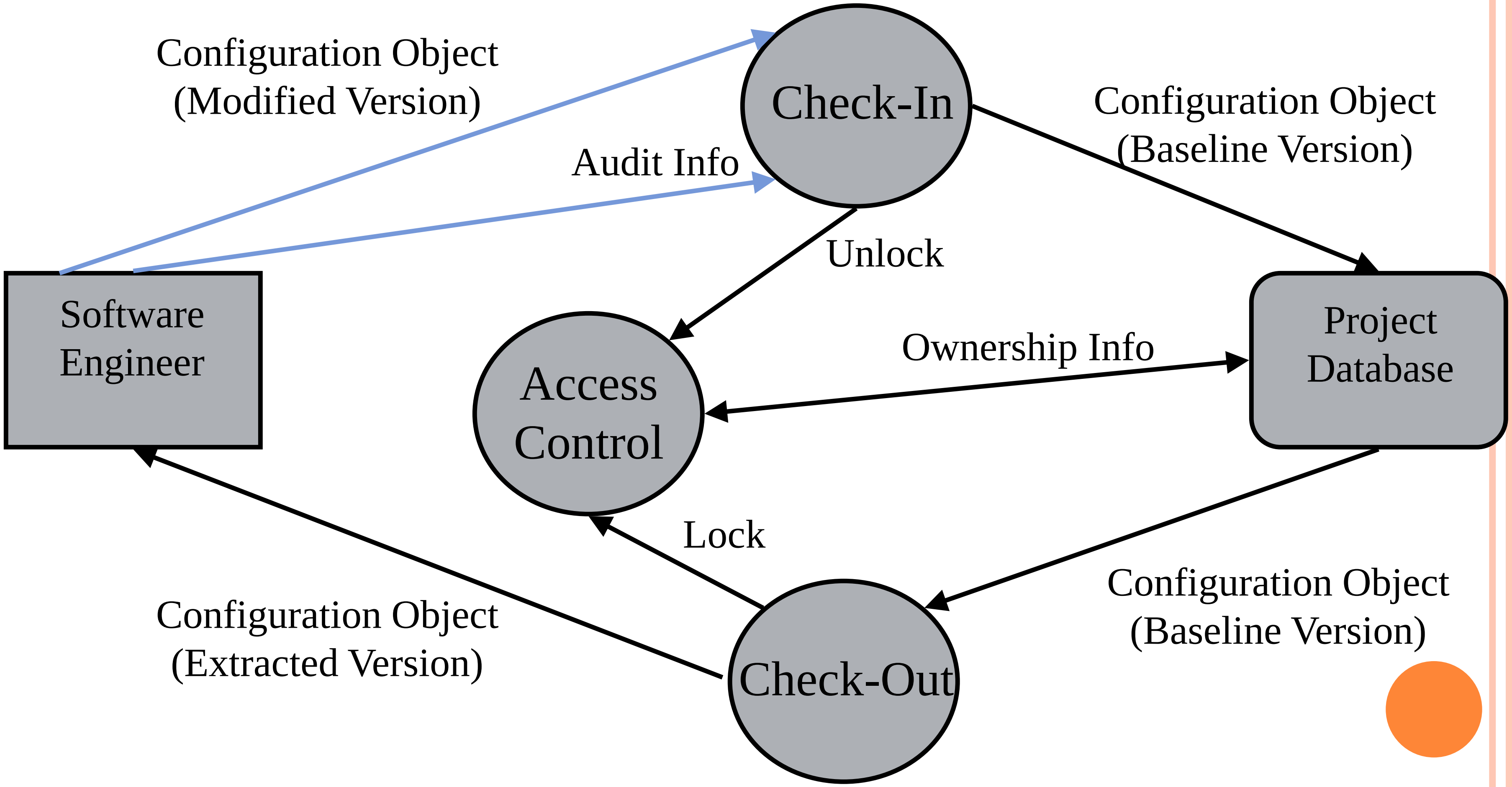
ACCESS AND SYNCHRONIZATION CONTROL



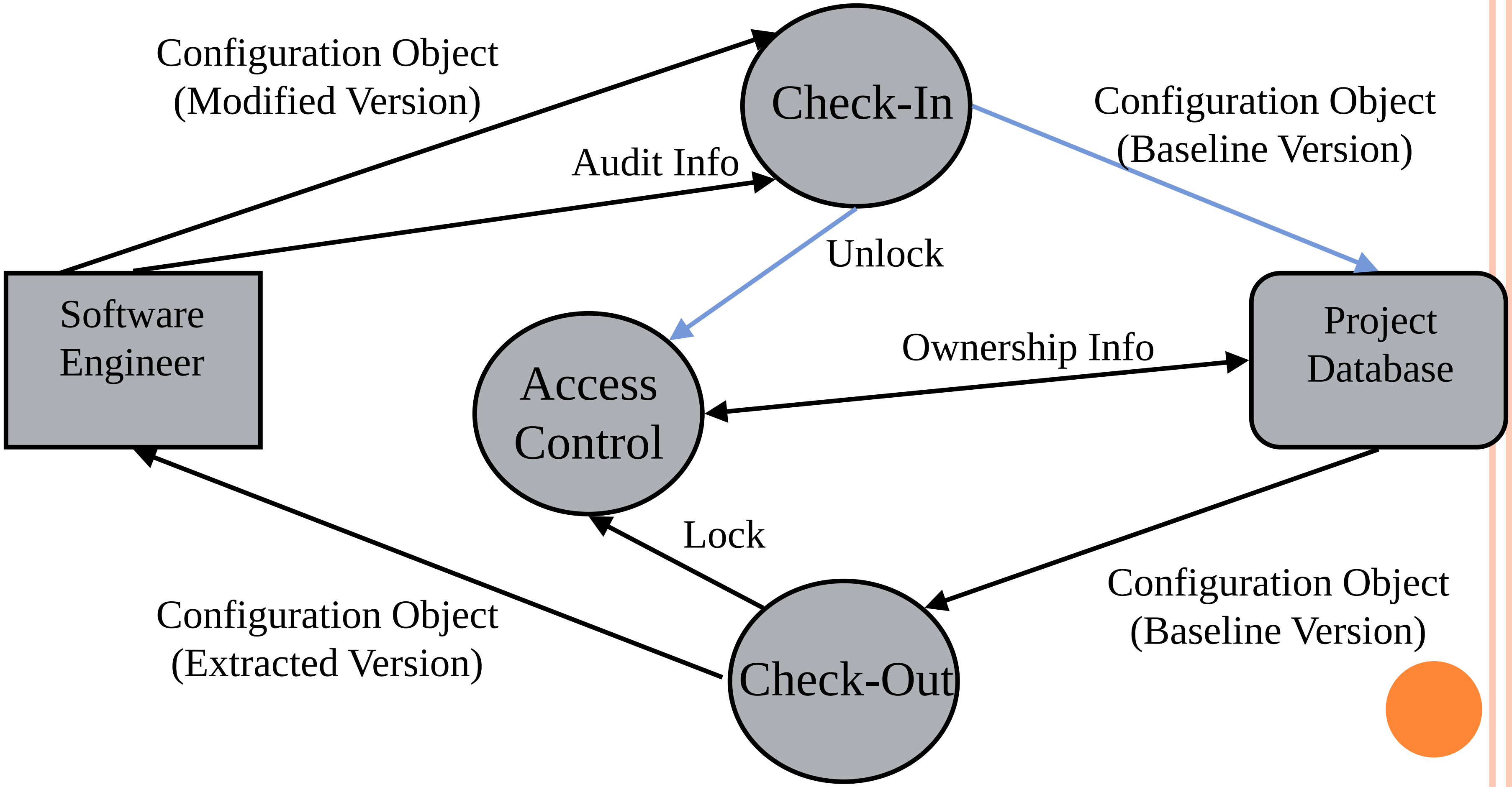
Access and Synchronization Control



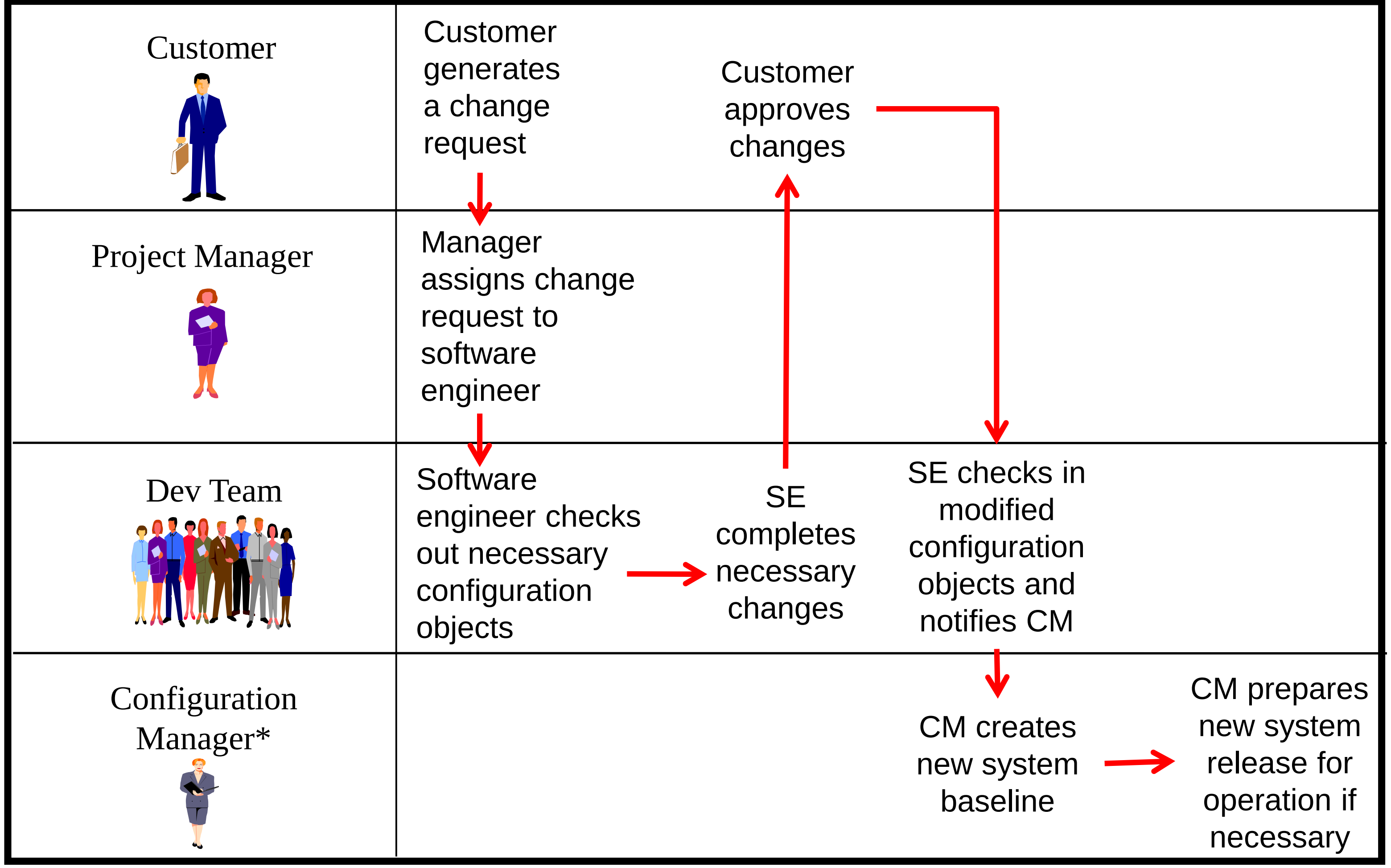
Access and Synchronization Control



ACCESS AND SYNCHRONIZATION CONTROL


Configuration Object (Modified Version)
Audit Info
Check-In
Configuration Object (Baseline Version)
Unlock
Software Engineer
Access Control
Ownership Info
Project Database
Lock
Configuration Object (Extracted Version)
Check-Out
Configuration Object (Baseline Version)

Configuration Management Cycle



Customer	Customer generates a change request	Customer approves changes	
Project Manager	Manager assigns change request to software engineer		
Dev Team	Software engineer checks out necessary configuration objects	SE completes necessary changes	SE checks in modified configuration objects and notifies CM
Configuration Manager*		CM creates new system baseline	CM prepares new system release for operation if necessary

*In charge of administering project database and providing access control to engineers

The SCM Process: **Software Configuration Auditing**

- Provides mechanism for determining the degree to which the current configuration of the software system mirrors the software system pictured in the baseline and the requirements documentation.

The SCM Process: **Software Configuration Auditing**

- Asks the following questions:
  - Has the specified change been made?
  - Has a formal technical review been conducted to assess technical correctness?
  - Has the software process been followed and standards been applied?
  - Have the SCM procedures for noting the change, recording it, and reporting it been followed?
  - Have all related SCIs been properly updated?

The SCM Process:
Software Configuration Status Reporting

- Provides a mechanism for maintaining a record of where the system is at any point with respect to what appears in published baseline documentation.

  - When a change proposal is approved it may take some time before the change is initiated or completed.

The SCM Process: **Software Configuration Status Reporting**

- Why needed?
  - Ensure that there is progress within the development of the project.
  - Track updates to baselines.

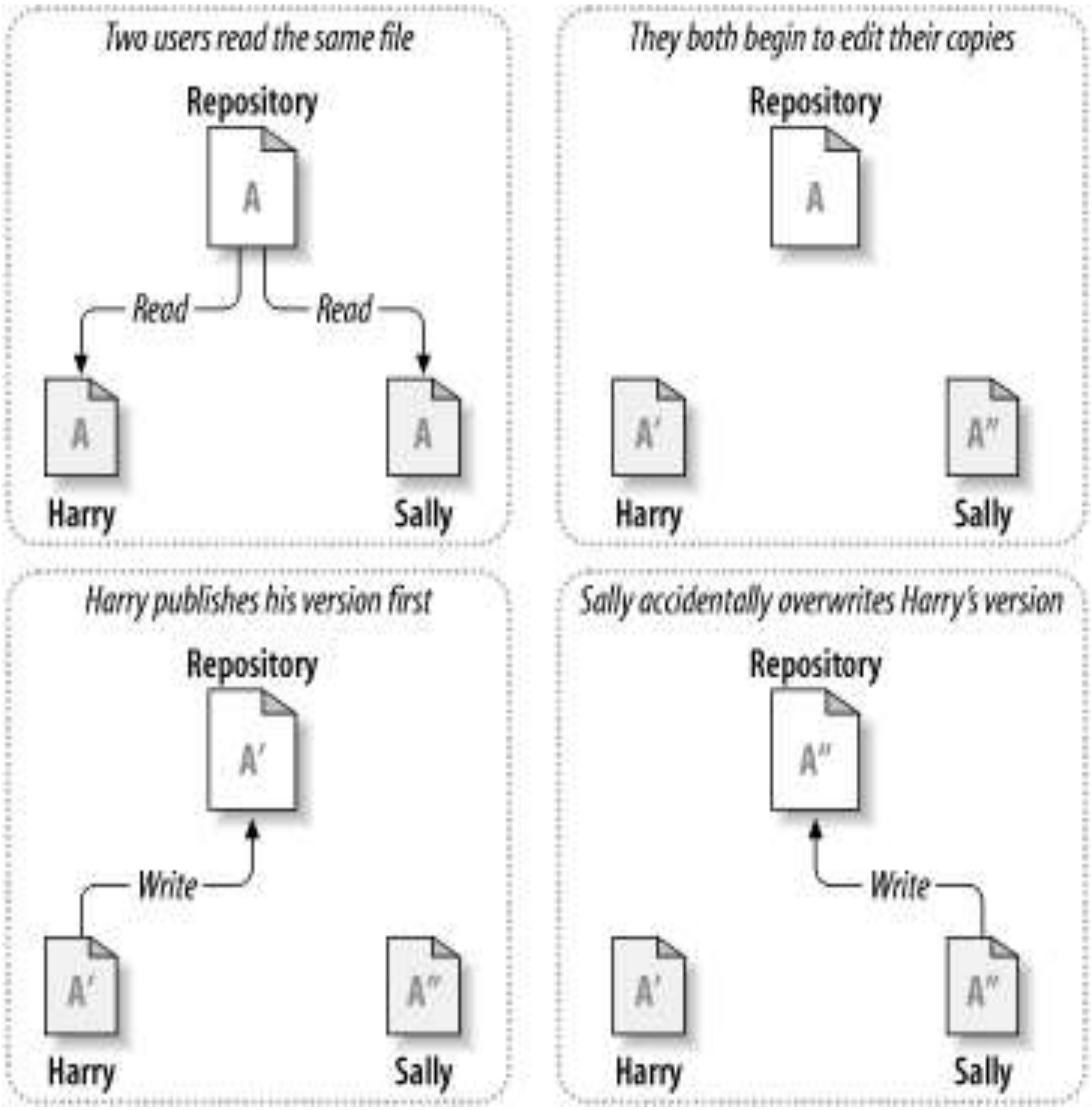
Requirements of CM

- Repository: shared DB for artifacts with controlled access to prevent overwrites.
- Version management: Maintain history of changes made to each artifact; provide ability to see how version was created.
- Work-space control: Private work space with ability to check out from repository and check in with new version number.
- Product modeling and building: Procedure to build the product from artifacts in repository.

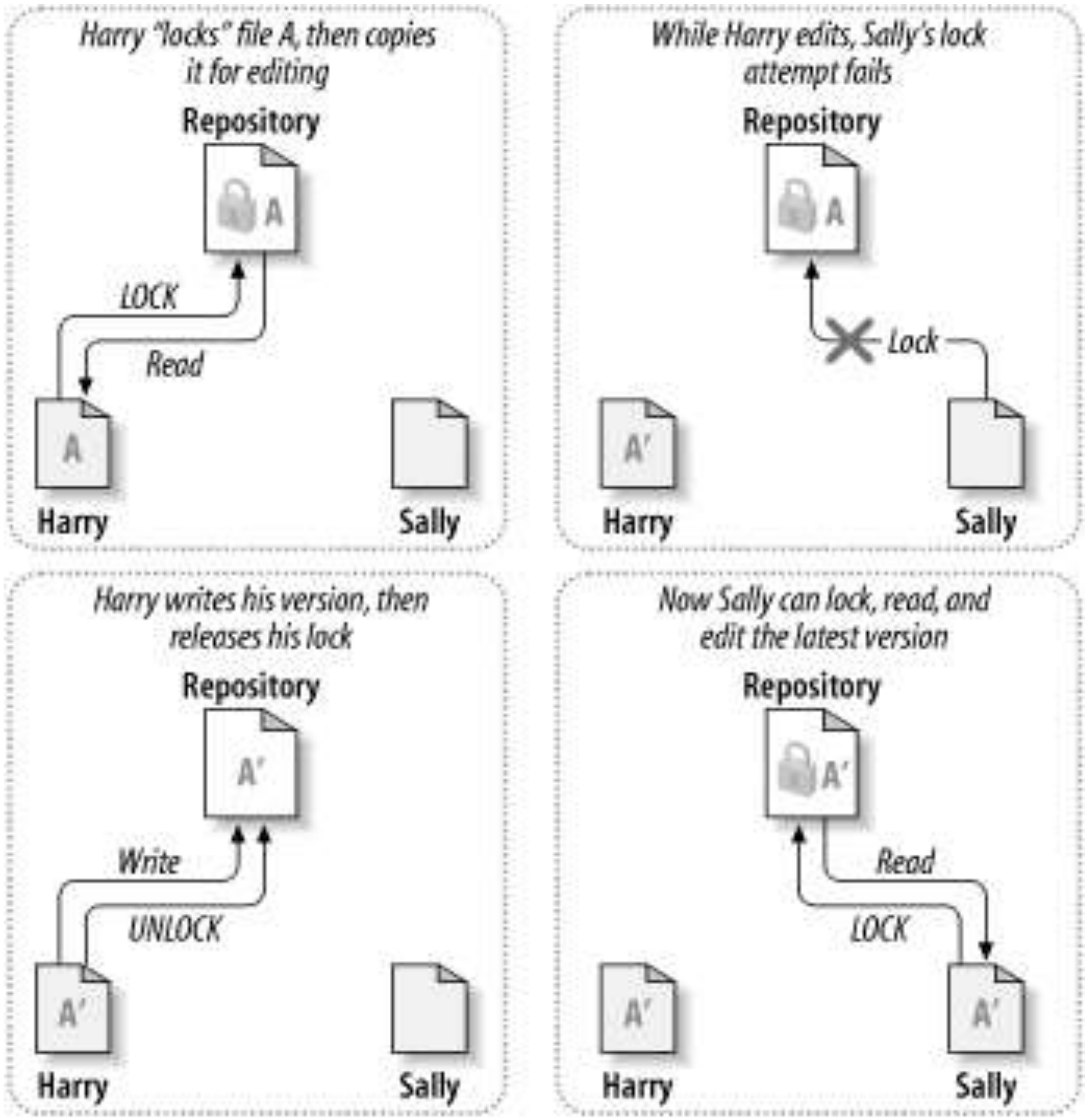
TOOLS FOR VERSION CONTROL

- The core mission of a version control system is to enable collaborative editing and sharing of data.
- File sharing is the most common problem faced by all version control systems, so most of the systems use Version Control with Subversion.
- For this the solutions can be:
  - Lock-Modify-Unlock Solution
  - Copy-Modify-Merge Solution

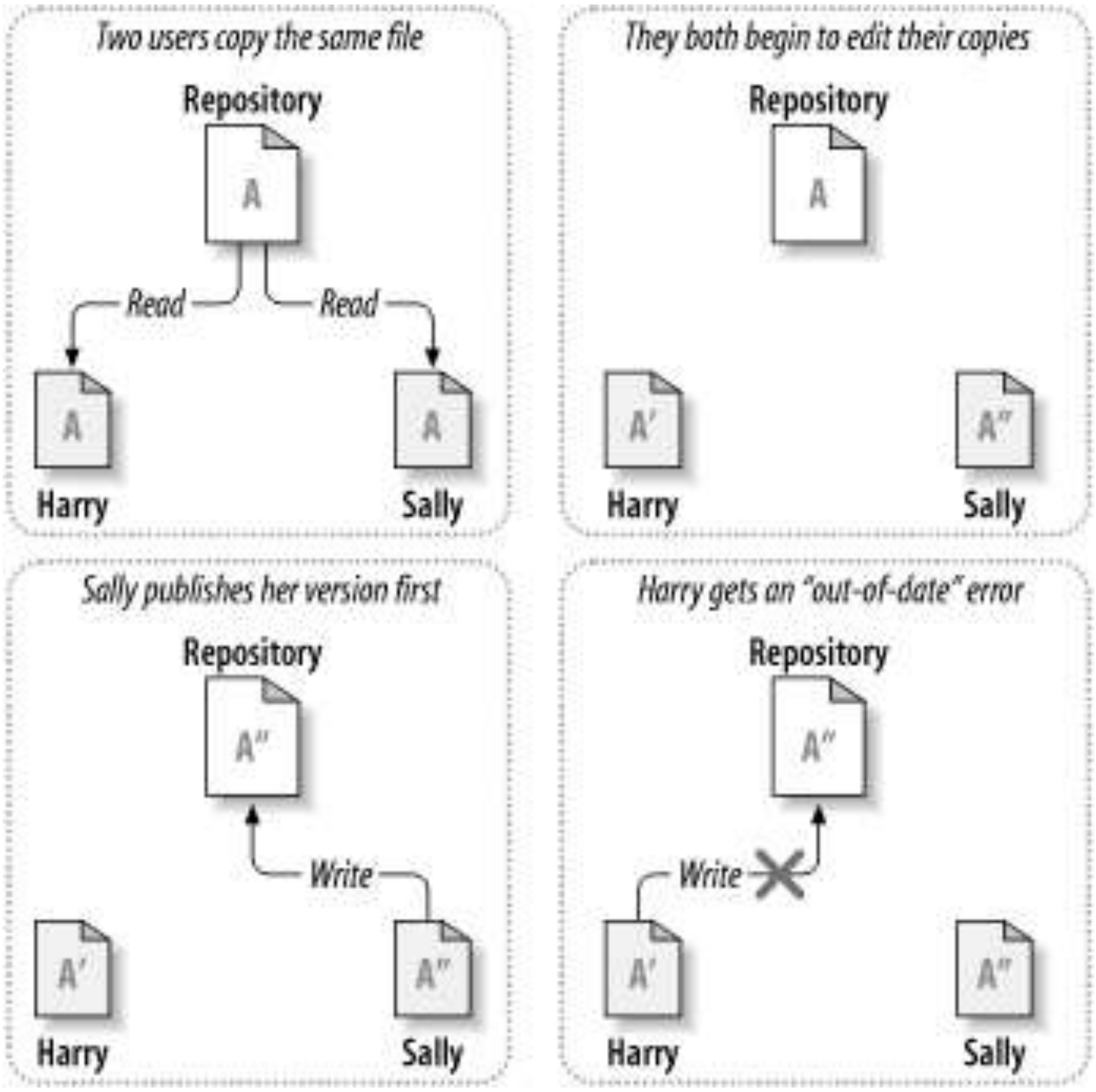
The Problem of File Sharing


Two users read the same file
Repository
A
Read
Read
A
A
Harry
Sally
They both begin to edit their copies
Repository
A
A′
A″
Harry
Sally
Harry publishes his version first
Repository
A′
Write
A′
A″
Harry
Sally
Sally accidentally overwrites Harry's version
Repository
A″
Write
A′
A″
Harry
Sally

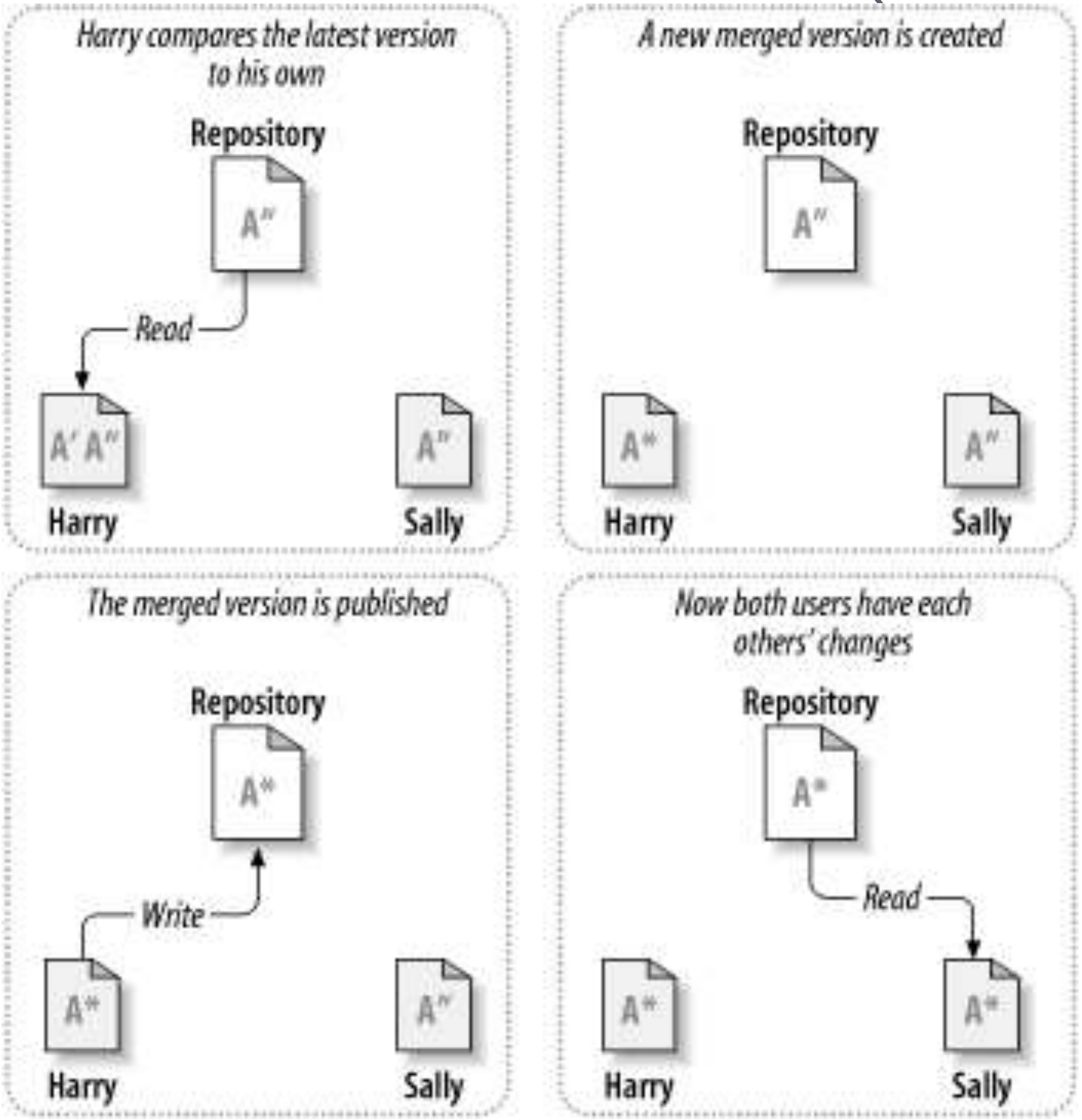
Lock-Modify-Unlock Solution


Harry "locks" file A, then copies it for editing
Repository
A
LOCK
Read
A
Harry
Sally
While Harry edits, Sally's lock attempt fails
Repository
A
Lock
A'
Harry
Sally
Harry writes his version, then releases his lock
Repository
A'
Write
UNLOCK
A'
Harry
Sally
Now Sally can lock, read, and edit the latest version
Repository
A'
Read
LOCK
A'
A'
Harry
Sally

Copy-Modify-Merge Solution


Two users copy the same file
Repository
A
Read
Read
A
A
Harry
Sally
They both begin to edit their copies
Repository
A
A′
A″
Harry
Sally
Sally publishes her version first
Repository
A″
Write
A′
A″
Harry
Sally
Harry gets an "out-of-date" error
Repository
A″
Write
A′
A″
Harry
Sally

Copy-Modify-Merge Solution (Cont...)


Harry compares the latest version to his own
Repository
A''
Read
A' A''
Harry
A''
Sally
A new merged version is created
Repository
A''
A*
Harry
A''
Sally
The merged version is published
Repository
A*
Write
A*
Harry
A''
Sally
Now both users have each others' changes
Repository
A*
Read
A*
Harry
A*
Sally