

Efficient Enterprise Builds With Apache Maven



Jason van Zyl
Sonatype



Importance of Infrastructure

It is usually assumed the infrastructure for a project is fine, but it is often the source of many problems because that assumption is false.

It is critical that the infrastructure for a project be sound in order for a project to succeed.





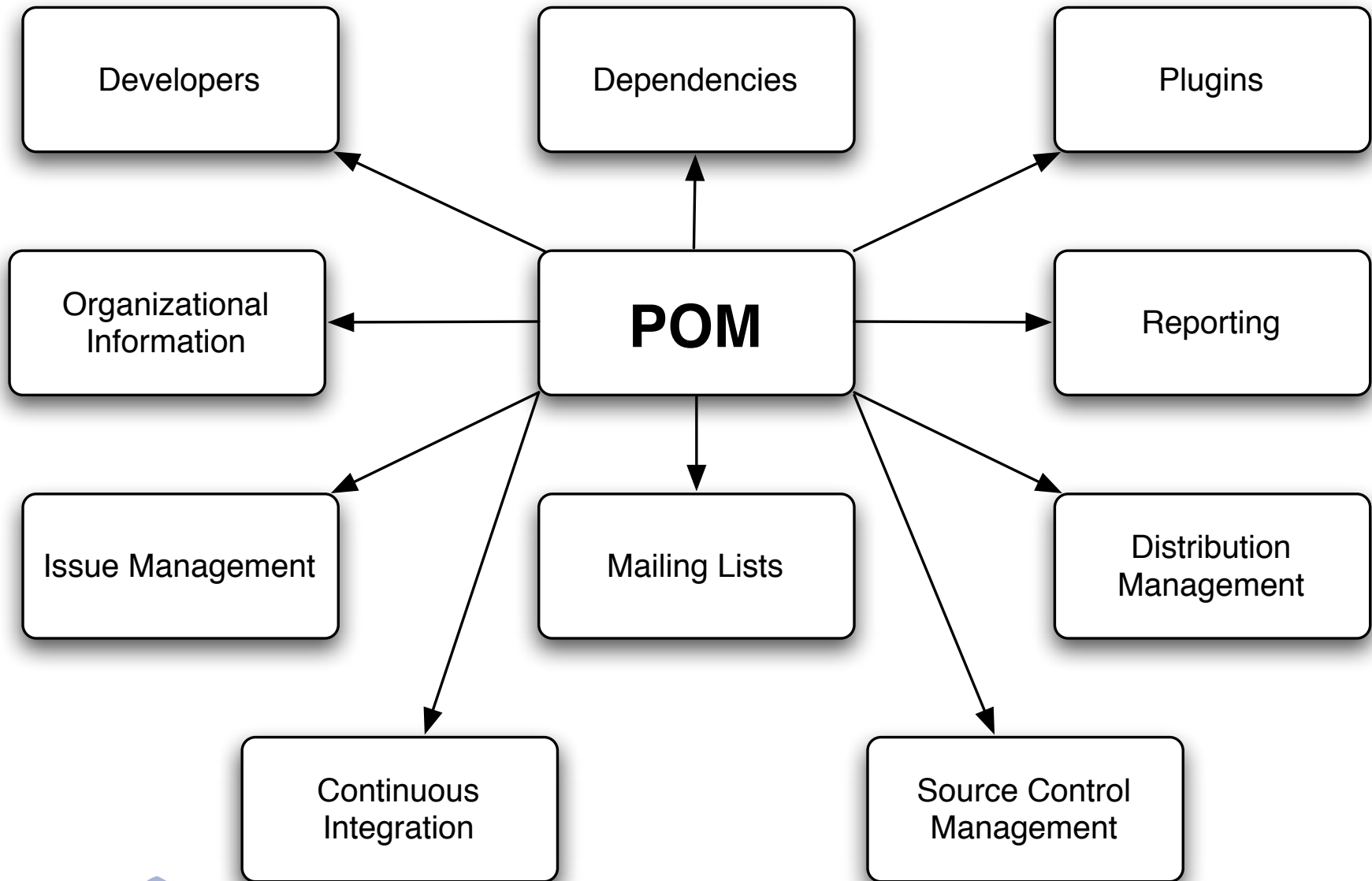
What is Maven?

- Build tool
 - Similar to Make, or Ant, but fundamentally different in that a larger context is considered
- Dependency management tool
 - Similar to Ivy, but not bolted on as an afterthought
- Site management tool



What maven really is ...

- Model for software projects
- Patterns for software development and development infrastructures
- Ultimately the basis for a new form of team collaboration

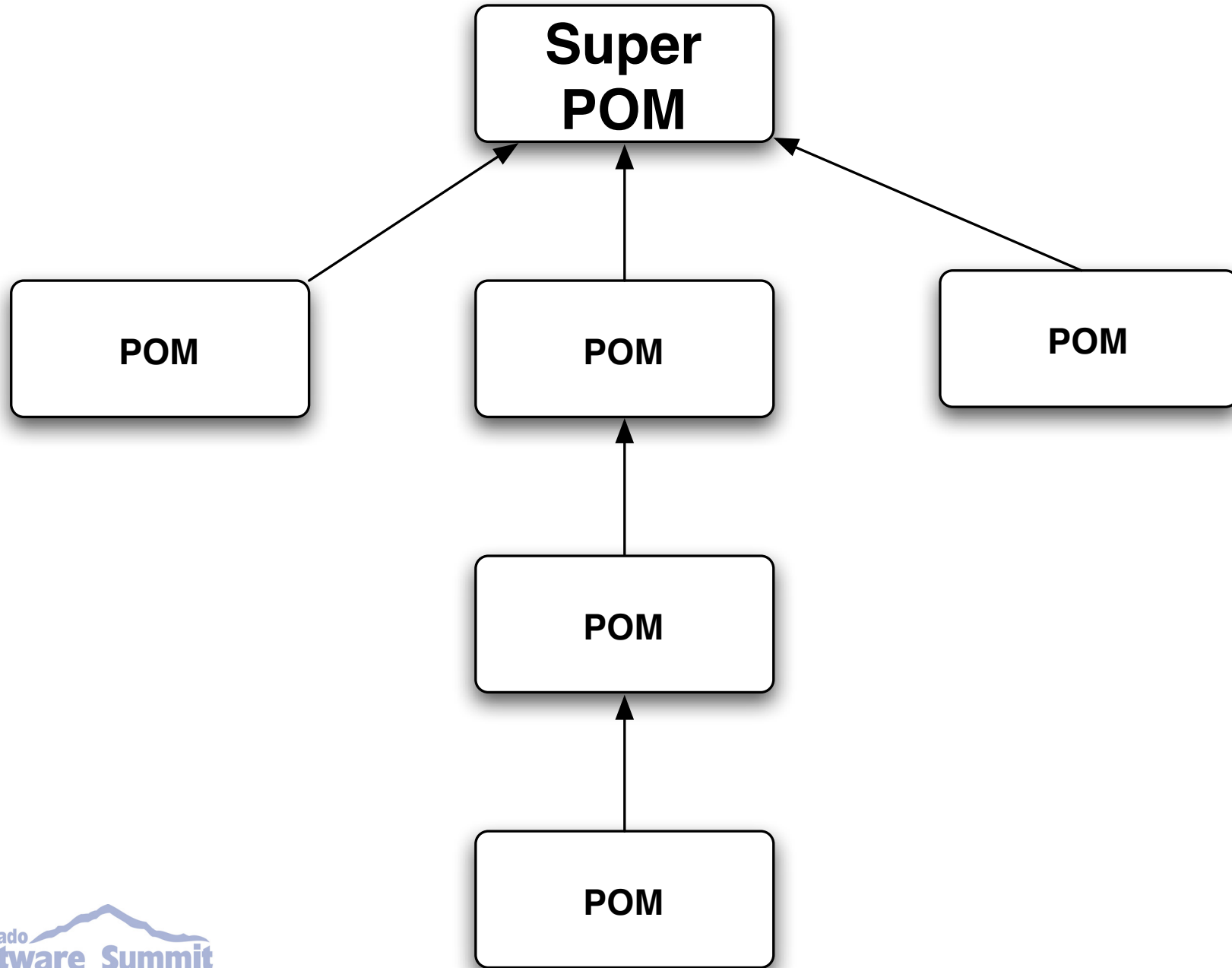


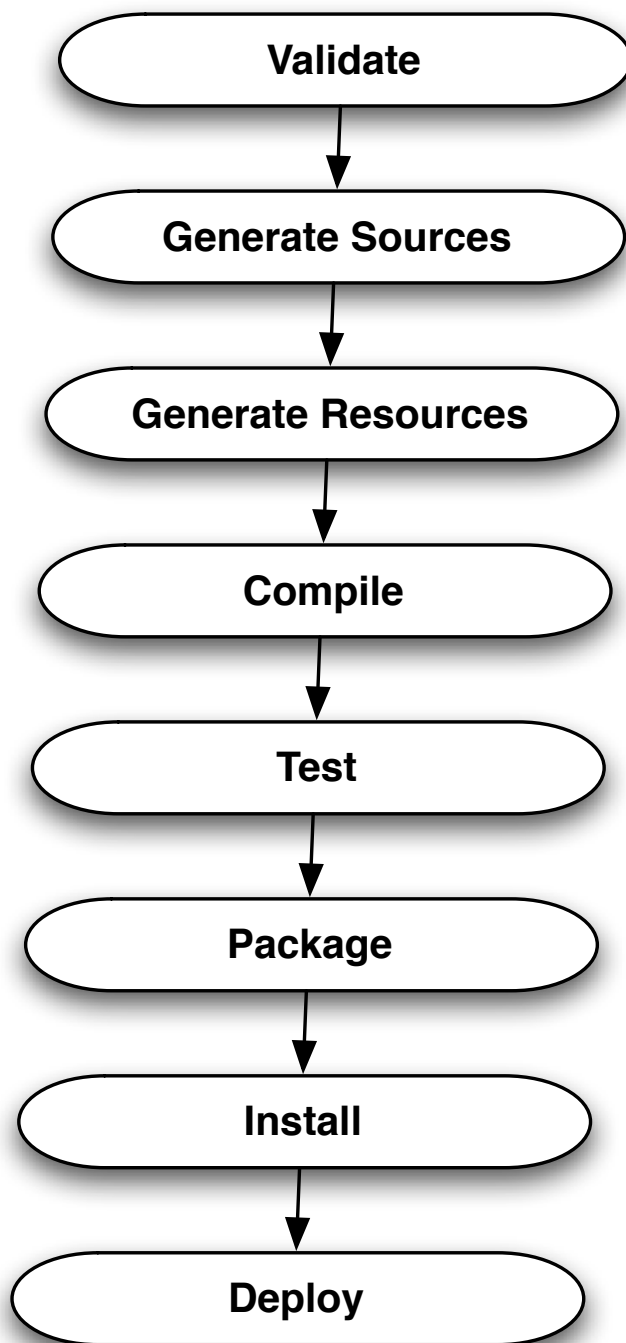


Simple Project Object Model

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sonatype.killerapp</groupId>
  <artifactId>superapp</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```









Development Infrastructures

- Development Infrastructures must provide facilities for:
 - Building
 - Testing
 - Documenting
 - Continuous Integration
 - Releasing
 - Artifact/Asset Management
 - Provisioning

A consistent model and patterns makes ...

- Automation easier
 - Releasing
 - Continuous Integration
 - IDE Bootstrapping
- Tooling easier
 - Dependency metadata
 - Plugin metadata
 - Standard lifecycle
- Real dependency analysis possible





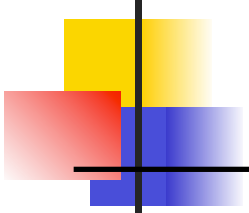
Why can't I just use Ant?

- You certainly can, but Ant provides no
 - conventions
 - models
 - processes
 - order

Infrastructure, Portland, Oregon and Maven

- Intentional infrastructure
 - You have to invest in your infrastructure to yield returns
 - It must be carefully planned, infrastructures don't just happen
- Long-term sustainability
 - The key is keeping people involved
 - sustainability by virtue versus sustainability by force
- Healthy growth in mind
 - Some constraints are necessary
- Promotion of community
 - Create opportunities for people to interact





Questions about infrastructure

- Do you think your infrastructure is wildly different than anyone else's?
- Is your infrastructure a competitive advantage?
- Would you like help with the most difficult infrastructure problems you might face?



Inherent utility

- The infrastructure should provide a lot to make complicated tasks easier
- When complicated tasks are made easy, there is more time for the bigger picture
- What's in the bigger picture?



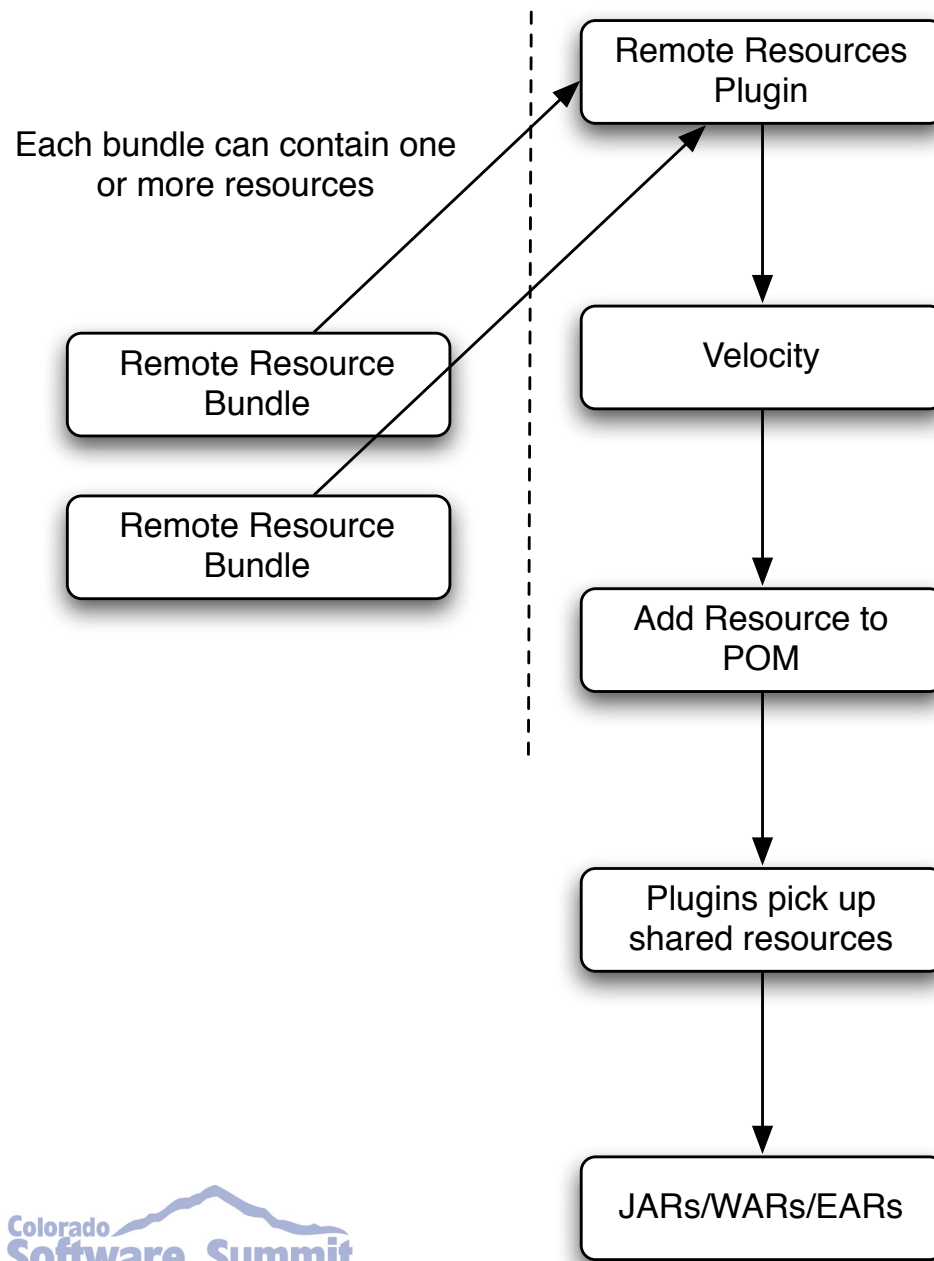
Maven & Social Capital

- Term coined by Jane Jacobs (*The Death and Life of Great American Cities*)
- People's ability to work together in groups
- Creation of real communities using the idea of patterns and conscious planning
- Concrete example of the Maven community working together to improve the release process at Apache



Apache Release Process

- Requirements
 - Apache License
 - Legal Notice
 - Disclaimer for Apache Incubator projects
 - PGP Signatures
 - Sources
 - Javadocs



The remote resources plugin takes the bundles, puts them in a ClassLoader and hands it off to Velocity.

Each resource from each bundle is processed through Velocity. The POM is pushed into the Velocity context.

All resources once processed are placed in `_${basedir}/target/maven-shared-archive-resources` and a Resource entry is added to the POM which corresponds to that directory.

Any archiving plugin that is aware of the shared archive resources Resource entry in the POM can pick them up and insert them into the archive it creates.

All archives created by projects using Maven now have their all their legal requirements for packaging satisfied transparently





Maven's Objectives

- Standards and Best Practices for build infrastructures
 - We use standard APIs and best practices when developing applications: Maven pushes this practice down to the level of infrastructure
- Provide a shared language for build infrastructure management
 - Patterns for build infrastructures in the spirit of *A Pattern Language* by Christopher Alexander
- Create healthy and robust build infrastructures that hold up to high degrees of flux



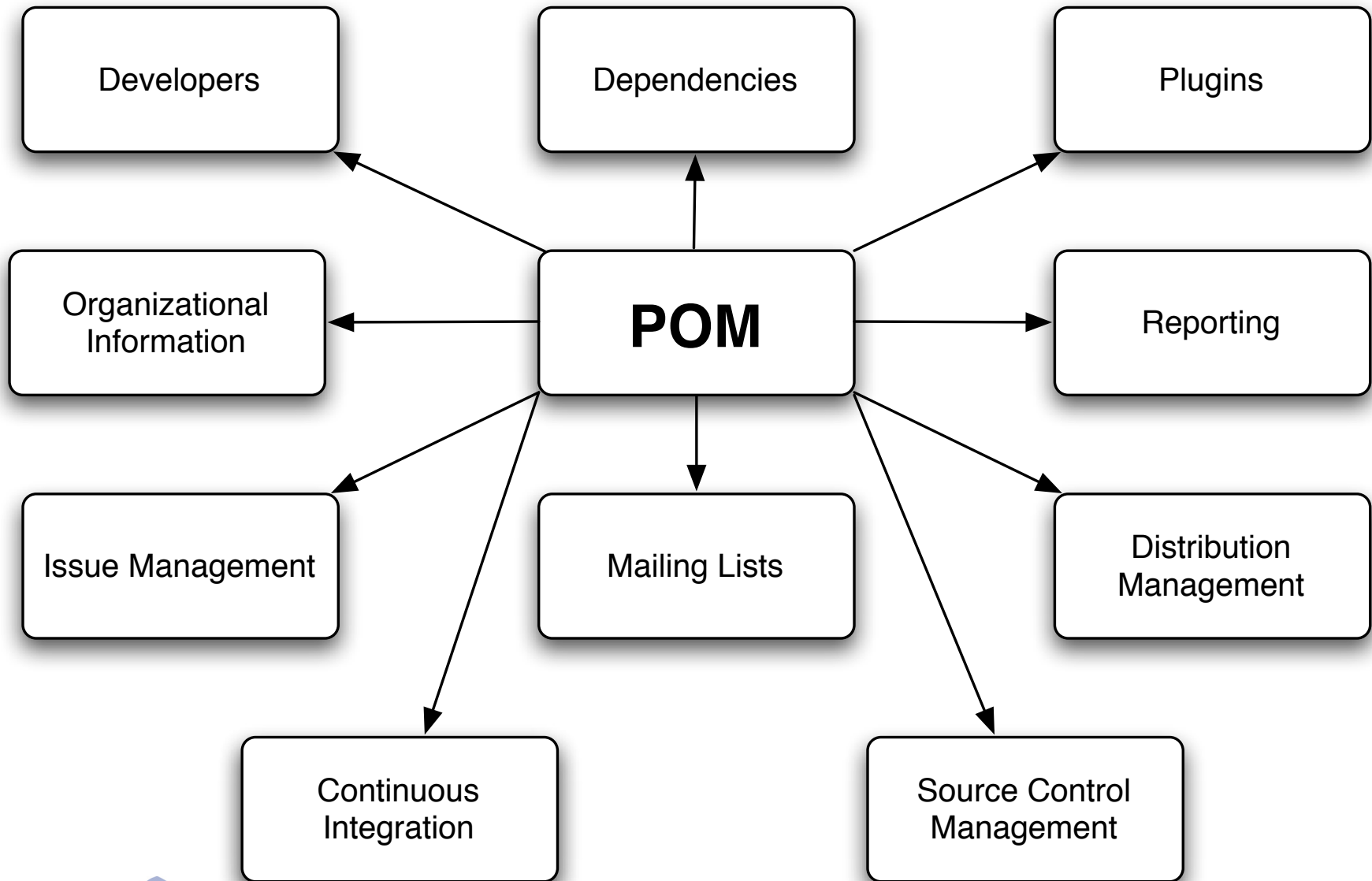
Maven's Principles

- Model driven development
- Convention over configuration
- Reuse and encapsulation of build logic
- Coherent organization of dependencies



Model driven development

- Analogous to MDA where your activities, actions and data model are governed by your architectural model, in Maven at the infrastructural level we try to provide a model
- Maven's project object model (POM)
- Maven's build life cycle

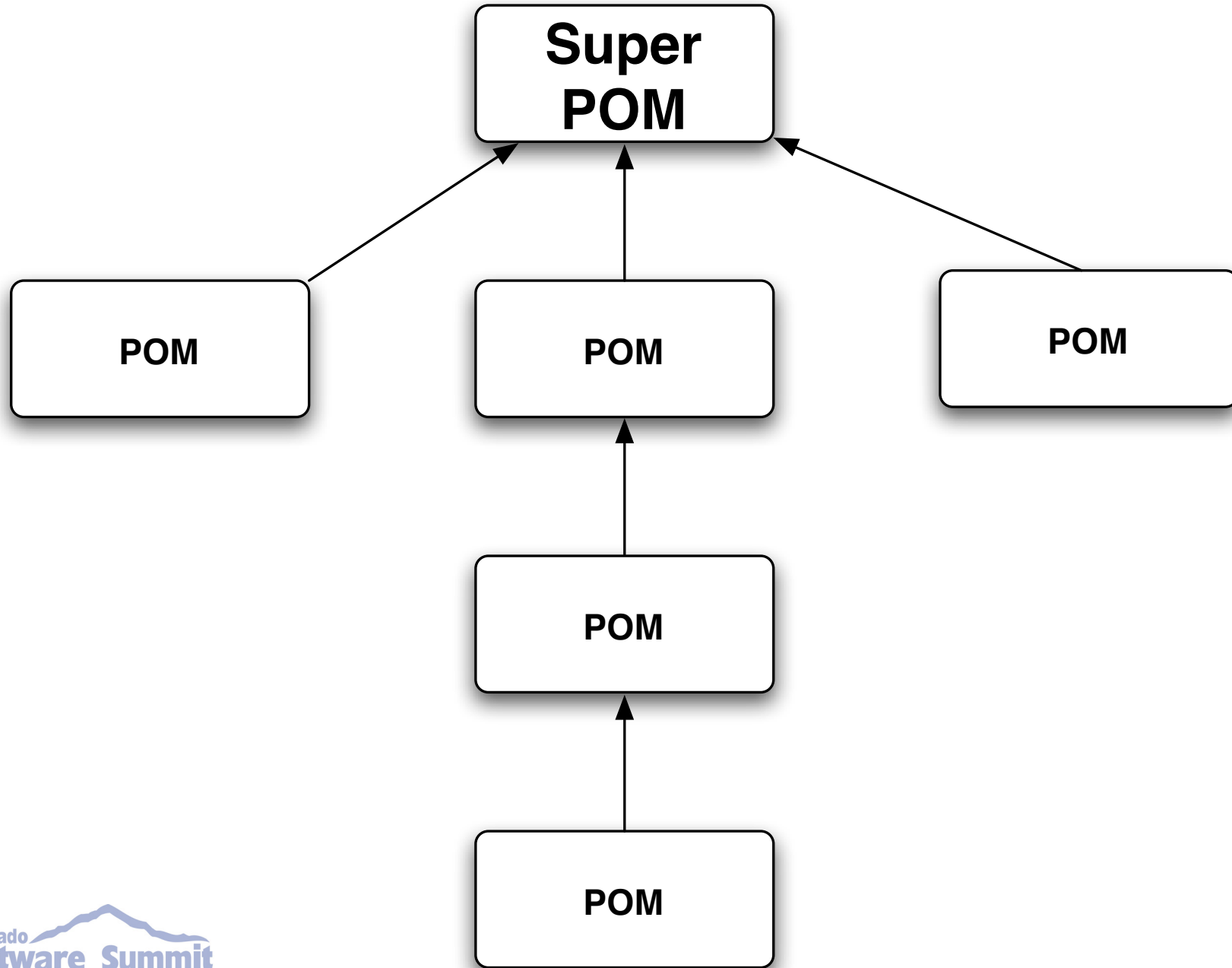


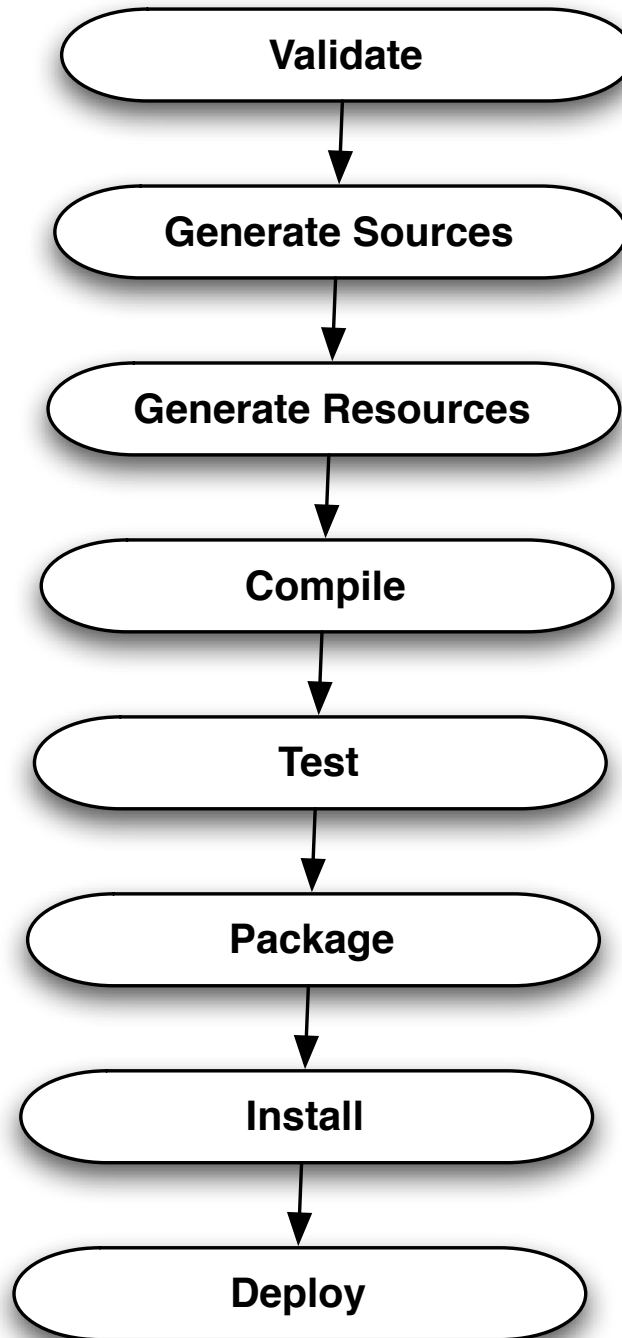


Simple Project Object Model

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.sonatype.killerapp</groupId>
  <artifactId>superapp</artifactId>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```









Convention over configuration

- Standard directory layout
- One primary artifact per build
- Standard naming conventions



Encapsulation and reuse of build logic

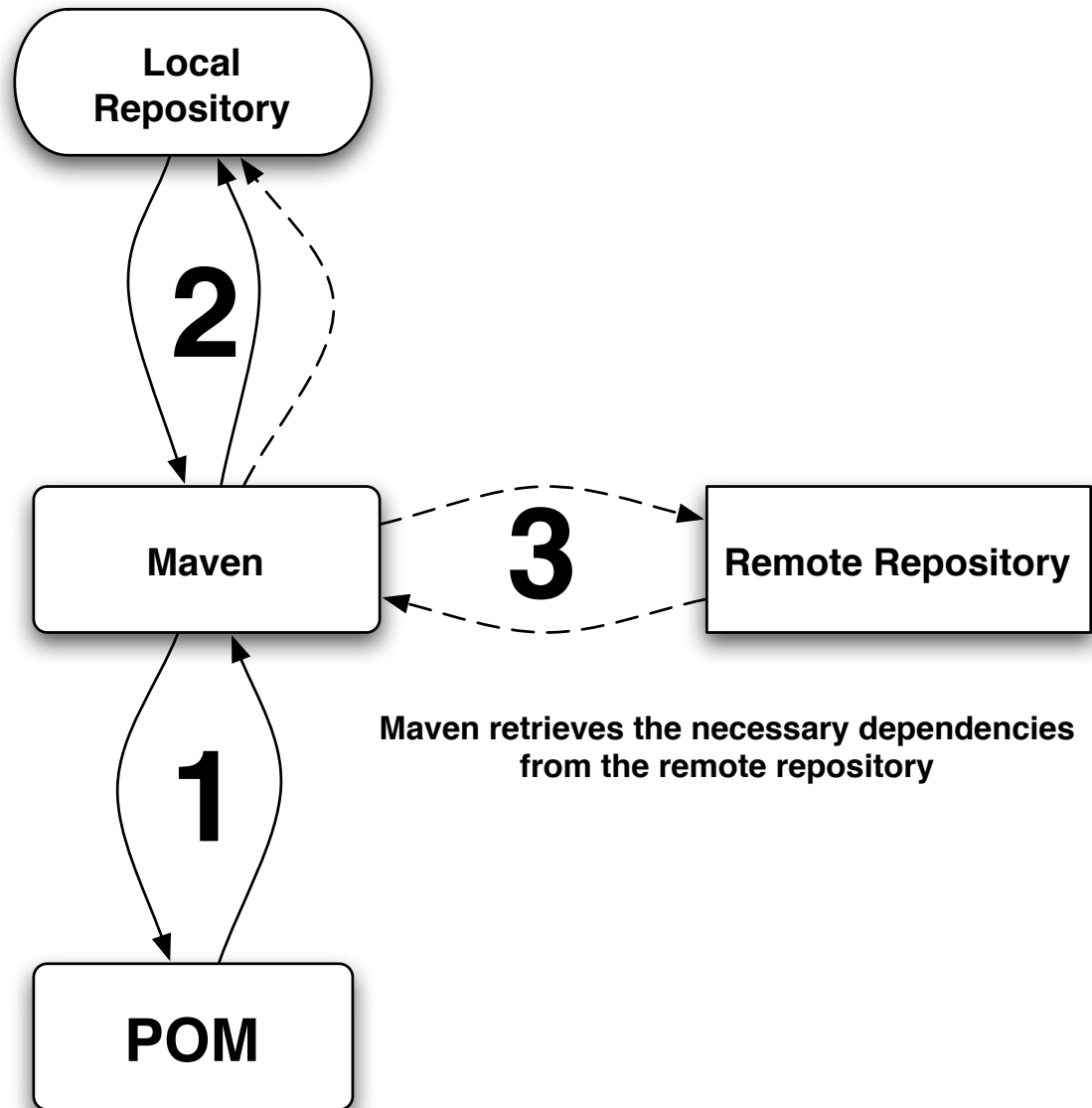
- All build logic is encapsulated in plugins
- Maven is a plugin execution framework
- Plugins can be applied to all Maven projects

Coherent organization of dependencies

- Say what you need, not where or how to get it
 - Dependencies in Maven are requested in a declarative fashion
- Artifacts and Repositories
 - Remote repositories for the satisfaction of dependencies
 - Local repository is the developers personal analog to a set of remote repositories



Maven checks the local repository for the required dependencies



Maven retrieves the necessary dependencies from the remote repository

Maven gathers dependency information from the POM





What Maven Provides

- Coherence – An orderly, logical, and aesthetically consistent relation of parts
- Reusability – Reuse not only of software components but the best practices of an entire industry
- Agility – Maven allows easier integration and project straddling
- Maintainability – Maven projects are more maintainable as there are far fewer surprises



Benefits of Maven

- Relieves the burden of project and build maintenance
- Easy for new users to embrace Best Practices
- Focus on adding value to your applications
- Draw upon the community for solutions



Hands on Demo!

- We will now convert an Ant project and conceptually map all the aspects of an Ant build to the Maven equivalent so you can see how powerful Maven actually is!



Questions?

- Do have any questions about Maven?