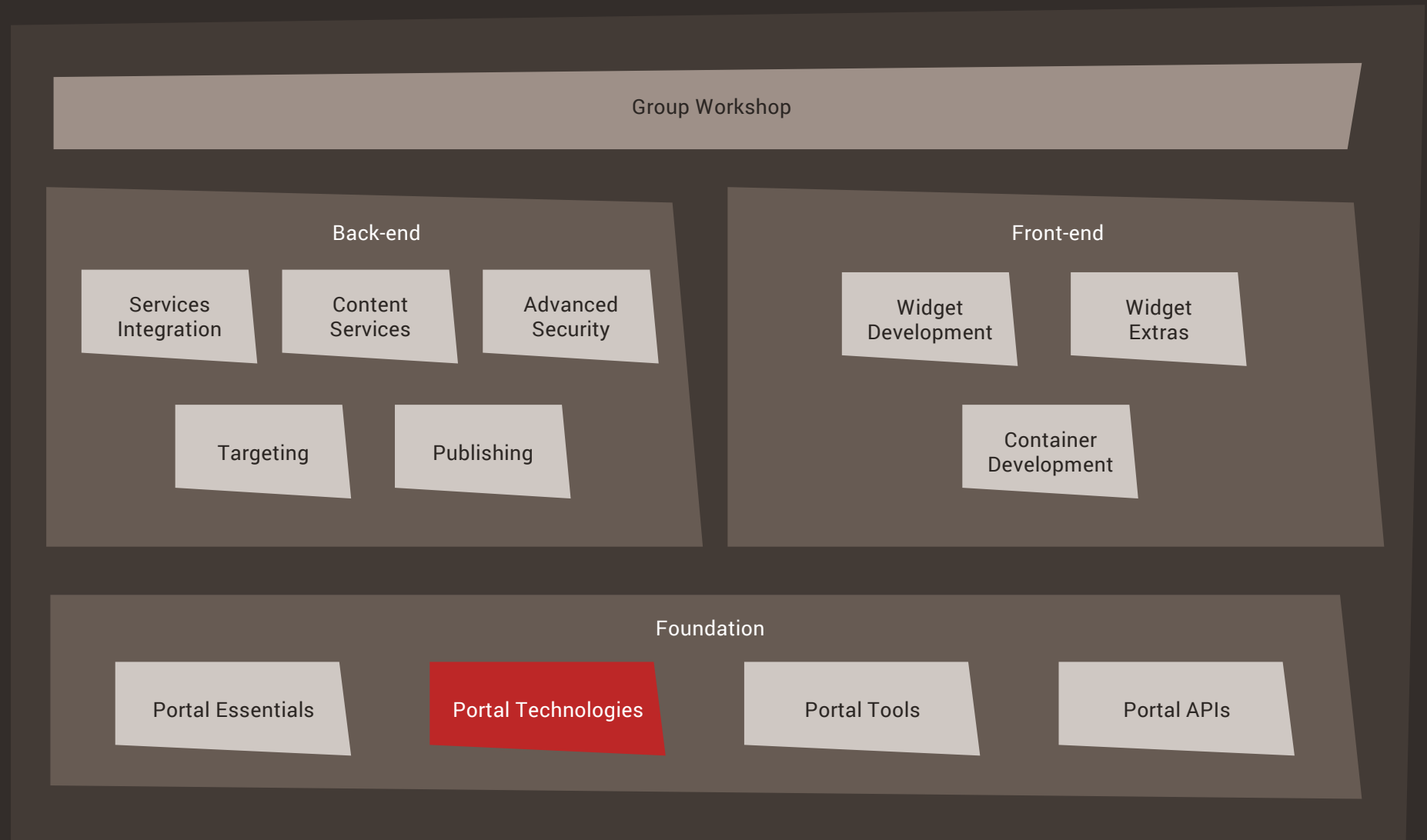




# Portal Technologies

## Focus Area: Foundation

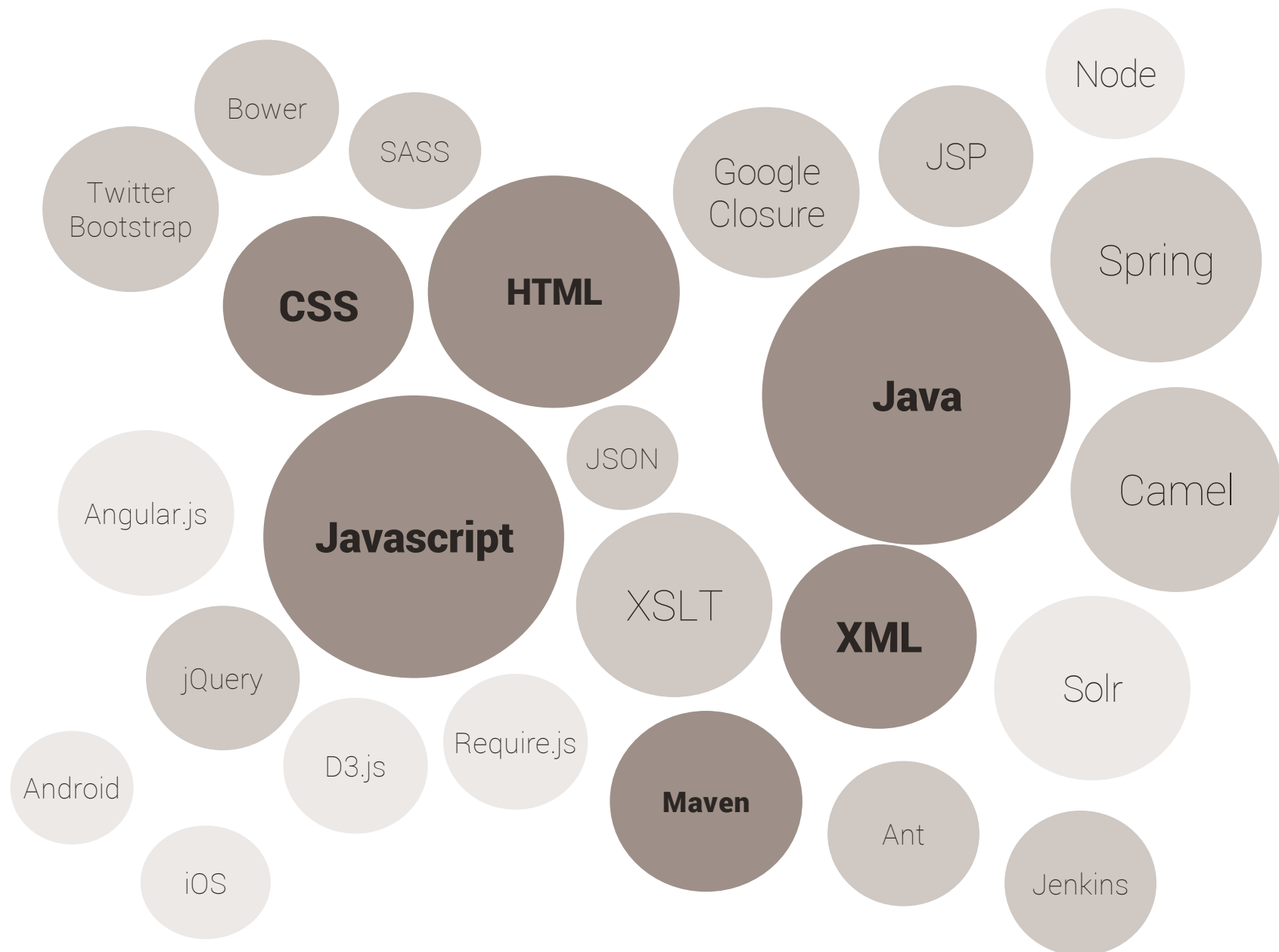




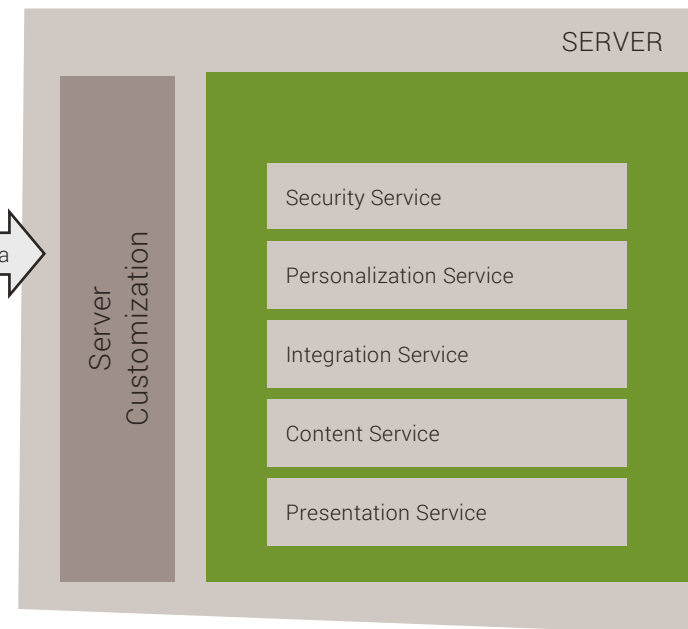
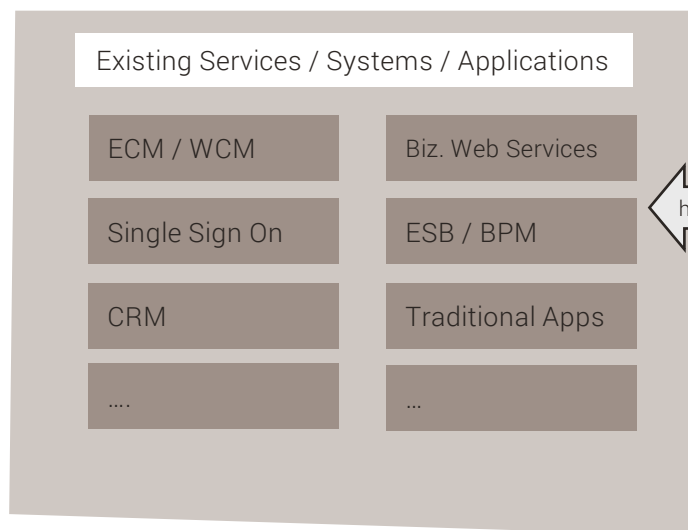
1. Get to know the technologies Portal is using under the hood
2. Get to know how to install and run a Portal server in your development environment
3. Become familiar with the development setup's directories and files

# Languages and Standards

Portal Technologies



## Client side developer



## Existing applications developer



## Java backend developer





# Technologies Overview

Portal Technologies

- Portal is an Enterprise Java Application
- Makes heavy use of Spring under the hood
- Key area's of Spring
  - Spring IOC
  - Spring MVC
  - Spring Security



- Portal client can render the model client side with JS
- Widget behavior is usually written with JS
- jQuery is shipped with Portal as a Portal Client dependency
  - But not a widget dependency

- Very ubiquitous throughout Portal
- Most configuration
  - Web app
  - Spring
  - Maven
- REST data interchange format

- Software tool for software project management and build automation.
- Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.
- Convention over configuration
  - `/src/main/java`
  - `/src/main/webapp`
  - `/src/test`

- Archetype is a Maven project templating toolkit
  - An original pattern or model from which all other things of the same kind are made.
- Artifacts
  - An artifact is a file, (usually a JAR), that gets deployed to a Maven repository.
  - Each artifact has a group ID (usually a reversed domain name, like *com.example.foo*), an artifact ID (just a name), and a version string.
  - A project's dependencies are specified as artifacts.
  - Artifacts are stored in a repository
    - Central repository
    - Local repository

- The Project Object Model (`pom.xml`)
  - Project meta data
  - Dependencies
  - Build lifecycle
    - Plugins

- Build lifecycle

process-resources	resources:resources
compile	compiler:compile
process-test-resources	resources:testResources
test-compile	compiler:testCompile
test	surefire:test
package	jar:jar
install	install:install
deploy	deploy:deploy

- Backbase hosts a repository that provides a convenient mechanism to setup a local Maven development environment.
- <https://repo.backbase.com/repo>
- Archetypes available from this repository:
  - Backbase Portal Archetype Blank
  - Launchpad Archetype



- Jetty is lightweight HTTP server and servlet container

- `maven-jetty-plugin`

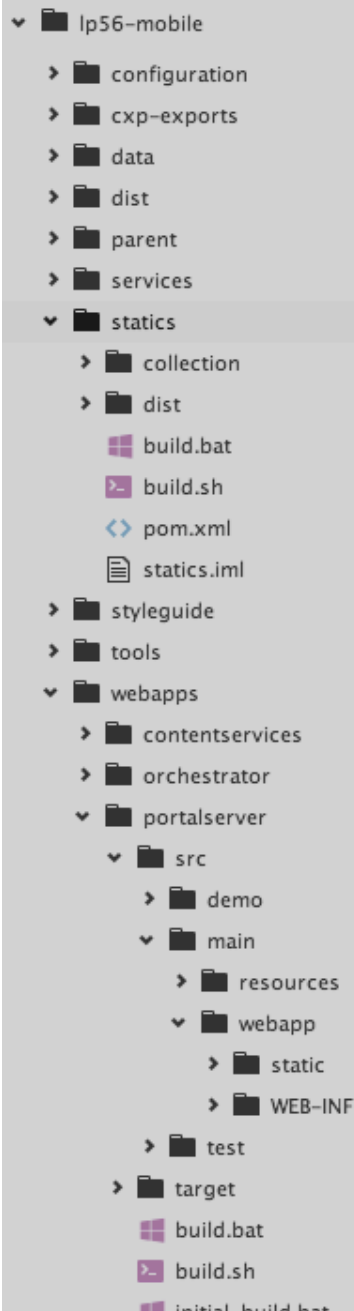
- Run using command

**`mvn jetty:run`**

- If you want your data to be reset to initial data then before `jetty:run` first execute the command:

**`mvn clean install`** (blank archetype)

**`mvn clean -Pclean-database`** (launchpad archetype)



## Highlights

- **webapps/** : Portal, CS, Solr & Orchestrator
- **configuration/** : all Backbase-specific configuration
- **services/** : custom Camel services
- **statics/** : Launchpad collection & custom items
- **styleguide/** : documentation of Launchpad Components

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-ns
  xsi:schemaLocation="
    http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.backbase.training</groupId>
  <artifactId>blank</artifactId>
  <version>1</version>
  <packaging>war</packaging>

  <name>blank</name>

  <repositories>[]

  <pluginRepositories>[]

  <properties>[]

  <dependencies>[]

  <build>[]
</project>
```



## Installing a CXP Development Environment

### Launchpad-based Project

<https://my.backbase.com/docs/how-to-guides/article//getting-your-first-launchpad-based-portal-set-up>

### Widget Collections 2.x Project

<https://my.backbase.com/docs/how-to-guides/article//setting-up-cxp-5-6-x-and-widget-collection-2-1-x-from-scratch>

# Archetype Structure

Portal Technologies

- Configuration: `/configuration`
- Source and test: `/src/main`, `/src/test`
  - Resources and config: `/src/main/resources`
    - Spring config: `/src/main/resources/META-INF/spring`
    - Portal import scripts: `/src/main/resources/import`
  - Java code: `/src/main/java`
  - Webapp code: `/src/main/webapp`
    - Static files: `/src/main/webapp/static`
    - WEB-INF: `/src/main/webapp/WEB-INF`
- Maven build directory: `/target`

- POM (Project Object Model): `/pom.xml`
  - Dependencies
  - Build
  - Plugins
    - Jetty
- Web app config: `/src/main/webapp/WEB-INF/web.xml`
  - Links to Spring config



- Portal uses a core set of technologies, but can integrate with many more
- Maven is a convenient way to run a local Portal development environment

# Thank you!

[www.backbase.com](http://www.backbase.com)  
[sales-eu@backbase.com](mailto:sales-eu@backbase.com)

New York: +1 646 478 7538  
Amsterdam: +31 20 465 8888