

Product Architecture, Installation, and Deployment

Fundamentals of IdentityIQ Implementation
IdentityIQ

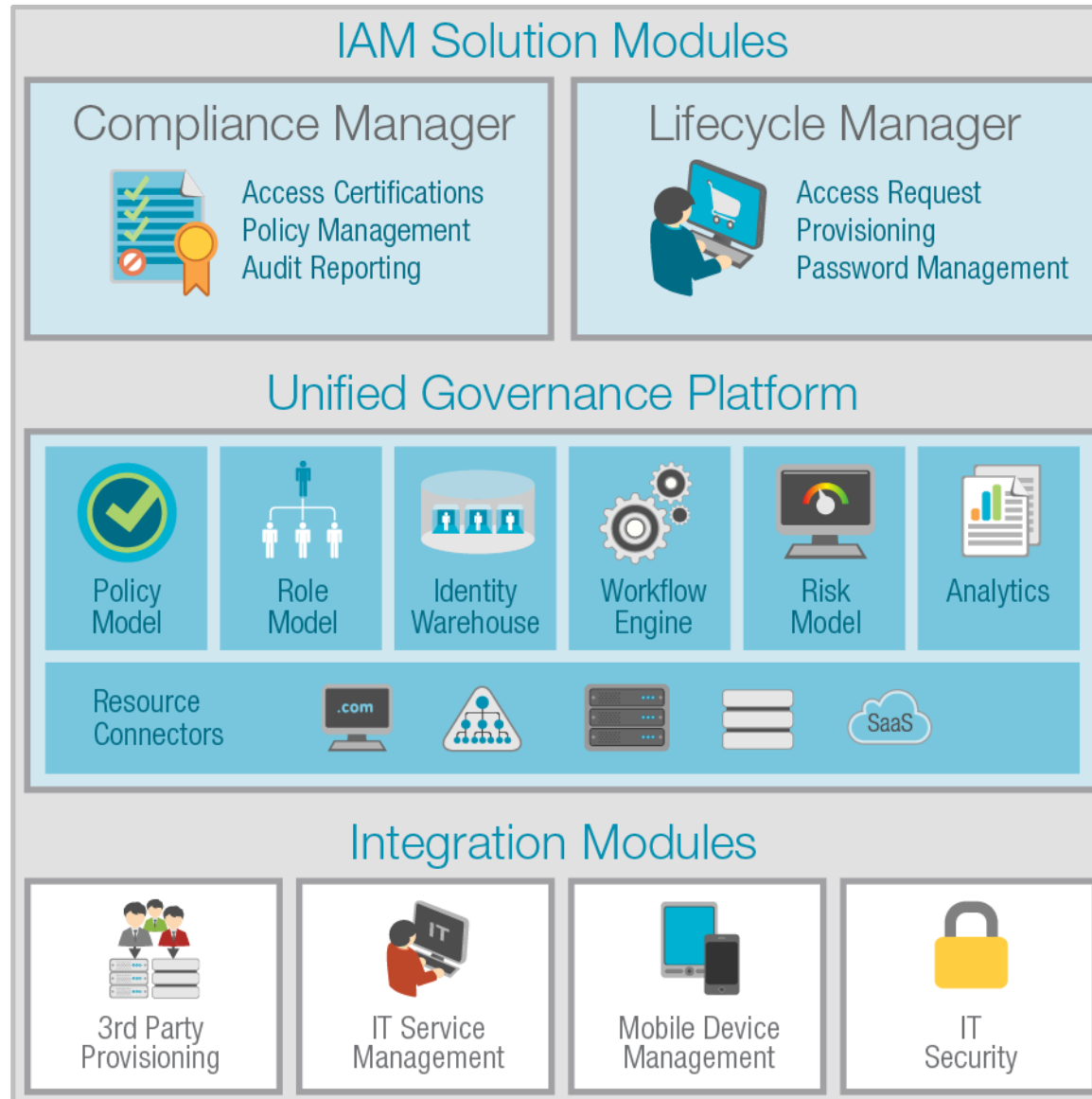
Overview

Product Architecture, Installation, and Deployment

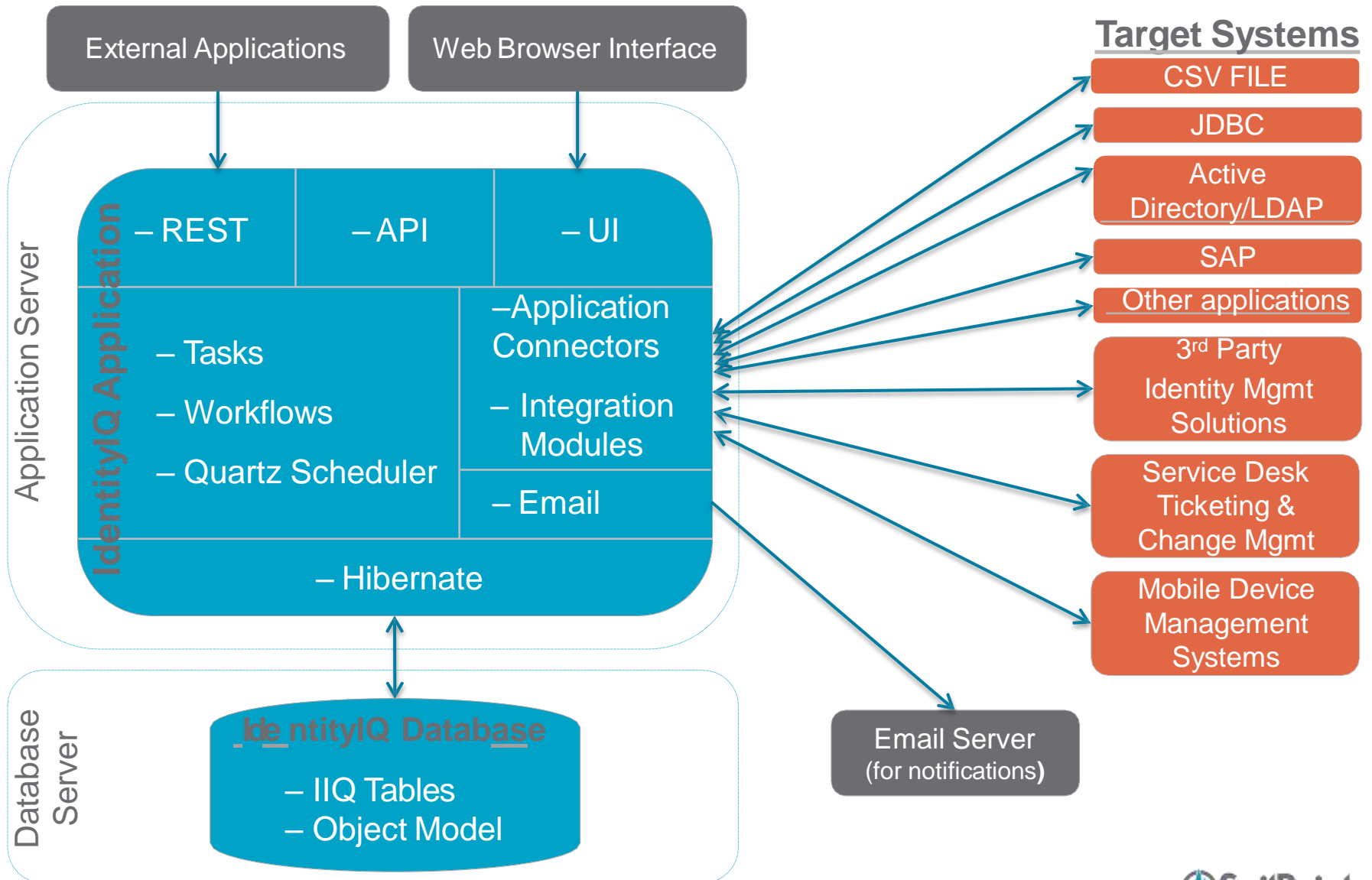
- Understand the Product Architecture
- Understand the Installation Procedure for IdentityIQ
- Understand how a Deployment Strategy and Environment Management can help with implementing IdentityIQ
- Understand Deployment Characteristics of IdentityIQ
 - Task Hosts and Request Hosts
 - Deployment Consideration for Database
 - High End Deployments (Redundancy)

ARCHITECTURE

IdentityIQ Product Components



Detailed Architecture Overview



Extension Levels

Extension Target	Method	Knowledge Needed
IdentityIQ Objects	Configuration <ul style="list-style-type: none">• Applications• Identity attributes• Rules• Etcetera	IdentityIQ Java XML
Web Application Objects	XHTML CSS Images	XHTML & JSF Web Design
Java	Compiled Code <ul style="list-style-type: none">• Custom tasks• Custom connectors• Workflow libraries• Connectivity	Java

INSTALLATION

IdentityIQ Installation

Overview

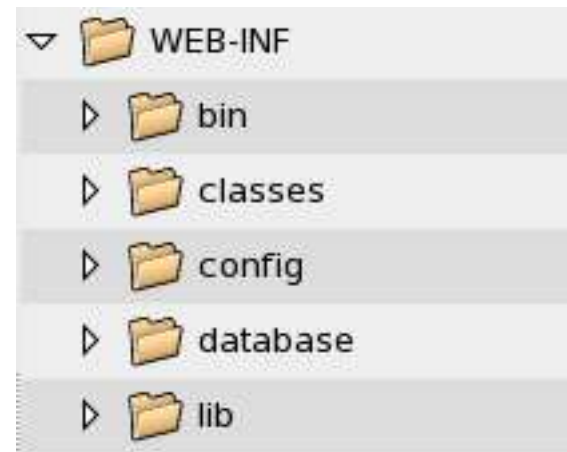
- Initial & Patch Deployment
 - Deploy WAR file
 - Modify and generate schema for IdentityIQ DB (Optional)
 - Prepare/Create IdentityIQ database
 - Initialize default system objects
 - Apply patches
- Ongoing Deployment & Operation
 - Initialize customized system objects
 - Deploy custom code
 - Deploy customized file-system artifacts

WAR File Deployment

- WAR (Web Application Archive)
 - File provided in product ZIP file
- Unzip or place WAR file into deployment directory of application server
 - WAR File Contents
 - Web Application Files – xhtml, html, CSS, images
 - Configuration Files – properties, xml
 - Docs – /docs directory
 - PDFs of product docs
 - Java Doc – for developers
 - Online Help

WEB-INF Directory

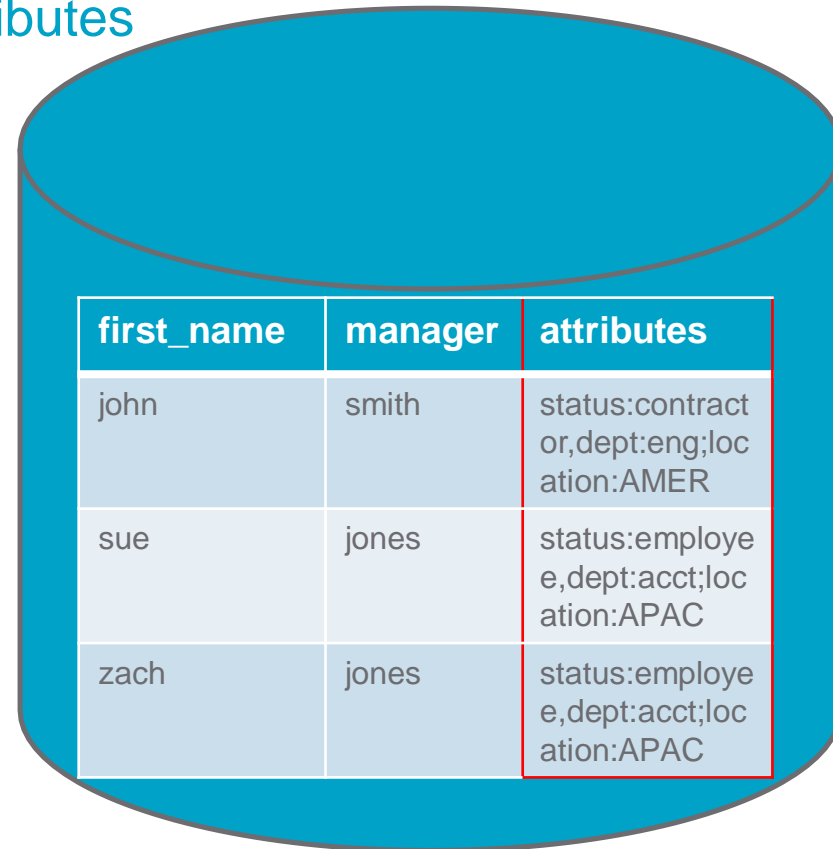
- \WEB-INF is an important directory within IdentityIQ
 - \WEB-INF\classes
 - Configuring IdentityIQ database connection properties
 - Configuring log4J
 - Configuring Database Searchable/Indexed attributes
 - \WEB-INF\bin
 - Running *iiq console*
 - Generating *iiq schema* files
 - Encrypting DB passwords
 - \WEB-INF\database
 - Database schema files
 - \WEB-INF\config
 - Files used to boot strap IdentityIQ
 - Example Files



IdentityIQ Database Configuration

Extended Attribute Definition

- Common to add business specific attributes
 - Called extended attributes
- Added through IdentityIQ GUI
- 6 objects can be extended
 - Applications
 - Roles (bundle)
 - Certifications
 - Identities
 - Accounts (link)
 - Entitlements (managed attributes)
- Default storage
 - Extended attributes are stored in a CLOB
 - No user database configuration is required
 - Efficient storage
 - *Not efficient for data that needs to be searchable*

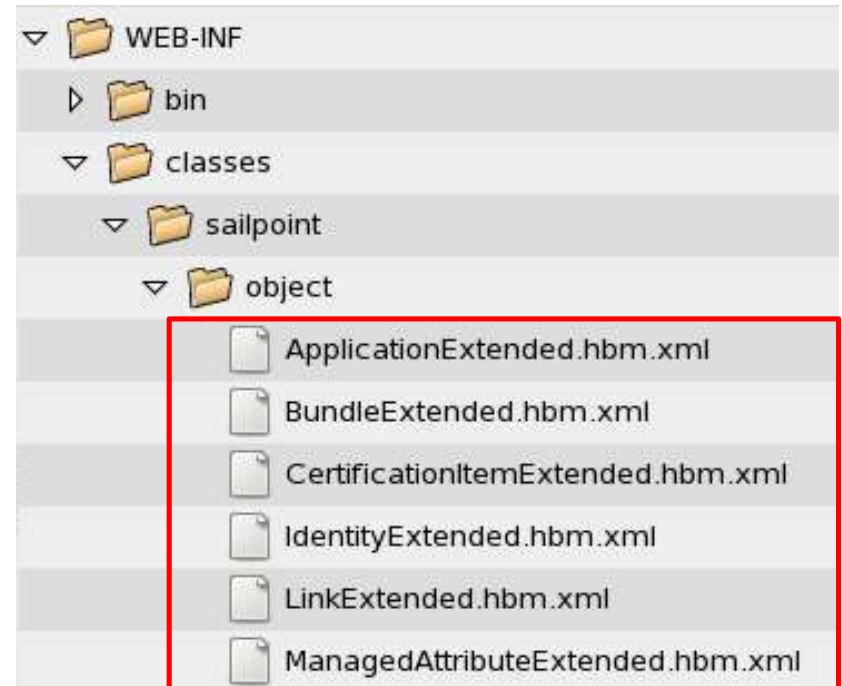


first_name	manager	attributes
john	smith	status:contractor,dept:eng;location:AMER
sue	jones	status:employee,dept:acct;location:APAC
zach	jones	status:employee,dept:acct;location:APAC

Database Schema Configuration

Configure Searchable Extended Attributes

- Create columns for extended attributes in IdentityIQ database
 - Edit the appropriate hibernate XML files
 - Generate schema
 - Generate database
- Add attributes to IdentityIQ and mark them as searchable



3 Types of Searchable Attributes

Configuring the Database

user_name	manager	...	department	location	extended_1	extended_2

Standard Attributes

*Pre defined by
IdentityIQ*

Named Extended Attributes

*Named column
defined by user*

Placeholder Extended Attributes

*Column space
defined by user*

Database Schema Configuration

Extending Searchable Attributes

- Named attributes

- Creates named column for attribute in IdentityIQ DB
- Mark extended attribute searchable in GUI and match to column
- Available objects
 - Identity, Application, Account, Role, Managed Attributes, Certification
- Unlimited (up to DB limits)

- Placeholder attributes

- Creates column with default name in IdentityIQ DB
- Mark extended attribute searchable in GUI, if no named match, will match to next available default column
- Available objects
 - Identity – default 10 searchable, 5 indexed
 - Identity to Identity – default 5, no indexed
 - Application – 4 extended attributes, 1 indexed
 - Account (Link) – 5 searchable attributes, 1 indexed
 - Role (Bundle) – 4 extended attributes, 1 indexed
 - Managed Attributes – 3 searchable, 3 indexed
 - Certification – 5 searchable attributes, 1 indexed
- Maximum 20 per object

Note: Be aware, indexing speeds up searching, but slows down updates

Database Schema Generation

- Console-based Schema Creation
 - Assures unique schema to each deployment
 - Command: **/WEB-INF/bin/iiq schema**
 - Generates DDL for all supported Databases
 - MySQL
 - Oracle
 - MS SQL Server
 - DB2
 - Filenames:
 - create_identityiq_tables.<databasetype>
 - example: create_identityiq_tables.mysql
 - Includes option for delta DDL
 - Command: **/WEB-INF/bin/iiq extendedSchema**

Database Preparation

■ Database Scripts

- Using your database tools of choice, create a database and all the necessary tables for IdentityIQ
- Scripts are provided
 - Out of the box (if you want to use the default schema)
 - Through console-based schema creation (customized schema)
 - For upgrade usage
- Location:
 - /WEB-INF/database
 - Examples:
 - create_identityiq_tables.mysql (custom)
 - create_identityiq_tables-8.0.mysql (default)
 - drop_identityiq_tables-8.0.mysql
 - upgrade_identityiq_tables.mysql
 - post_upgrade_identityiq_tables.mysql

Note: If generating custom scripts, take care to load proper files. Look at date/time stamps to make sure you are using the most recently generated files.

Configure IdentityIQ Properties

/WEB-INF/classes/iiq.properties

Data Source Properties

dataSource.maxWait=10000

dataSource.maxActive=50

dataSource.minIdle=5

#dataSource.minEvictableIdleTimeMillis=300000

#dataSource.maxOpenPreparedStatements=-1

dataSource.username=identityiq

dataSource.password=1:iCAlakm5CVUe7+Q6hVJIBA==

MySQL 5

dataSource.url=jdbc:mysql://localhost/identityiq?useServerPrepStmts=true&tinyInt1isBit=true&useUnicode=true&characterEncoding=utf8

dataSource.driverClassName=com.mysql.jdbc.Driver

sessionFactory.hibernateProperties.hibernate.dialect=sailpoint.persistence.MySQL5InnoDBDialect

Database Username

Database Password
Encrypt using *iiq encrypt*
command

Data Source URL
specifying
host/port/database

Initialize IdentityIQ Default Objects

- Newly created IdentityIQ database will be empty
- Initializing IdentityIQ will setup all System Objects
 - Out-of-the-box users, reports, default tasks, workflows, etc.
- Initializing IdentityIQ

```
/WEB-INF/bin/iiq console
> import init.xml
```
- Initializing IdentityIQ Lifecycle Manager

```
/WEB-INF/bin/iiq console
> import init-lcm.xml
```

Note: This process of loading an XML file is often used for your own deployment (for example, your applications, rules, roles, tasks, etc.)

Verify IdentityIQ Installation

- After IdentityIQ is installed and configured
 - Start the Application Server
 - Login to IdentityIQ
 - `http://<server>:<port>/identityiq/`
 - Default User:
 - `spadmin/admin`
 - Server can be deployed at the root of app server
 - Example: <http://server.domain.com/>

How to Reset an IdentityIQ Installation

- To reset system
 - Stop app server
 - Drop and recreate the database
 - From database console
 - > drop database identityIQ
 - > source <*your script here*>
 - Reload initialization files
 - From IdentityIQ console
 - > import init.xml
 - > import init-lcm.xml (if using Lifecycle Manager)
 - Start app server

DEPLOYMENT

Important Considerations

- App Server/DB Choices
- How many IdentityIQ environments?
 - Sandbox
 - Development
 - Quality assurance
 - Production
- Redundancy Requirements
- UI versus Batch Hosts
 - Small installations have one host that handles both UI requests and batch tasks
 - Larger installations may have a mixture of UI and Batch hosts
 - Batch handles
 - Workflows
 - Tasks/reports
 - Certification generation
 - UI hosts handles user interactions
 - Access Requests
 - Performing Certifications
 - Dynamic Analytics

System Choices

Supported Platforms

- Application Servers

- Tomcat
- WebSphere
- WebLogic
- JBoss

- Databases

- MySQL
- Oracle
- MS SQL Server
- DB2

- Java Platform

- Sun, Oracle or IBM JDK
- Oracle JRockit JDK

- Browsers

- Firefox ESR
- Internet Explorer
- Google Chrome
- Safari

- Mobile Support

- Safari for iPad
 - End user functionality
For example: approvals
and access certifications

- Deploy what you are most comfortable maintaining!!!!

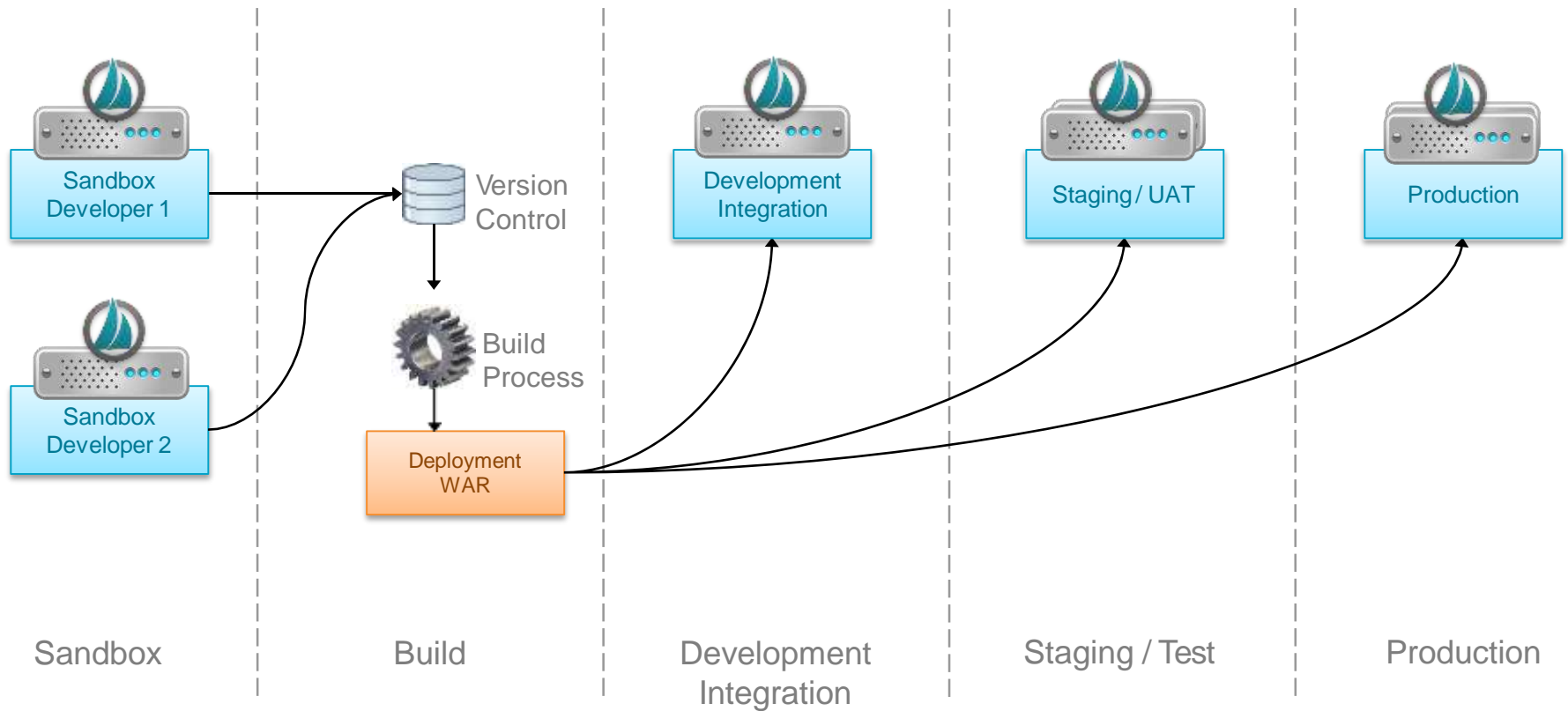
Deployment Strategy

Best Practice

- **Sandbox – Developer Environment**
 - Individual IdentityIQ system per developer
 - Typically limited memory, disk space and running in a VM
 - Load small amount of representative data
- **Development – Unit Test Environment**
 - System for multiple developers to test code together
 - Load small amount of representative data
- **Staging –Test Environment**
 - User acceptance, functional testing, etc.
 - Similar to production
 - Can be used for performance and stress testing
- **Production Environment**
 - Incorporates redundancy and failover

Deployment Strategy

Environment Management Best Practice



Build Process

Services Standard Build (SSB)

- Created and used by SailPoint Professional Services for deployment across multiple customer sites
- Automates the packaging and deployment of custom objects and code across all environments
- Build configuration for Apache Ant build tool
- Utilize directly or as a model for creating a build process

- SSB Process

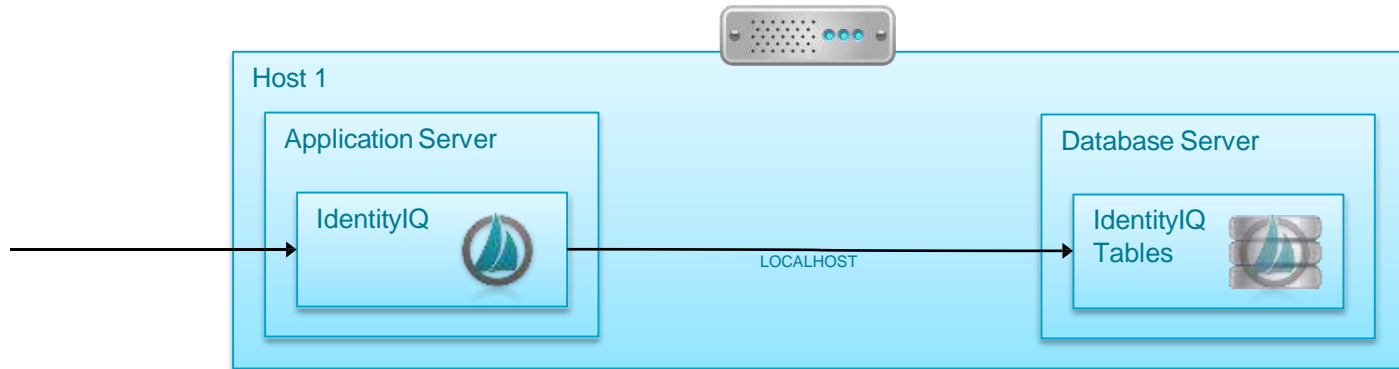
- Export objects from sandbox into XML files
- Push XML files to version control system
- Use the build tool to build *.war* from the version control directory
- Release packaged war to additional environments

Note: For dissimilar environments (for example, Windows for sandboxes and Linux for test and production) SSB supports token replacement

- Available on Compass

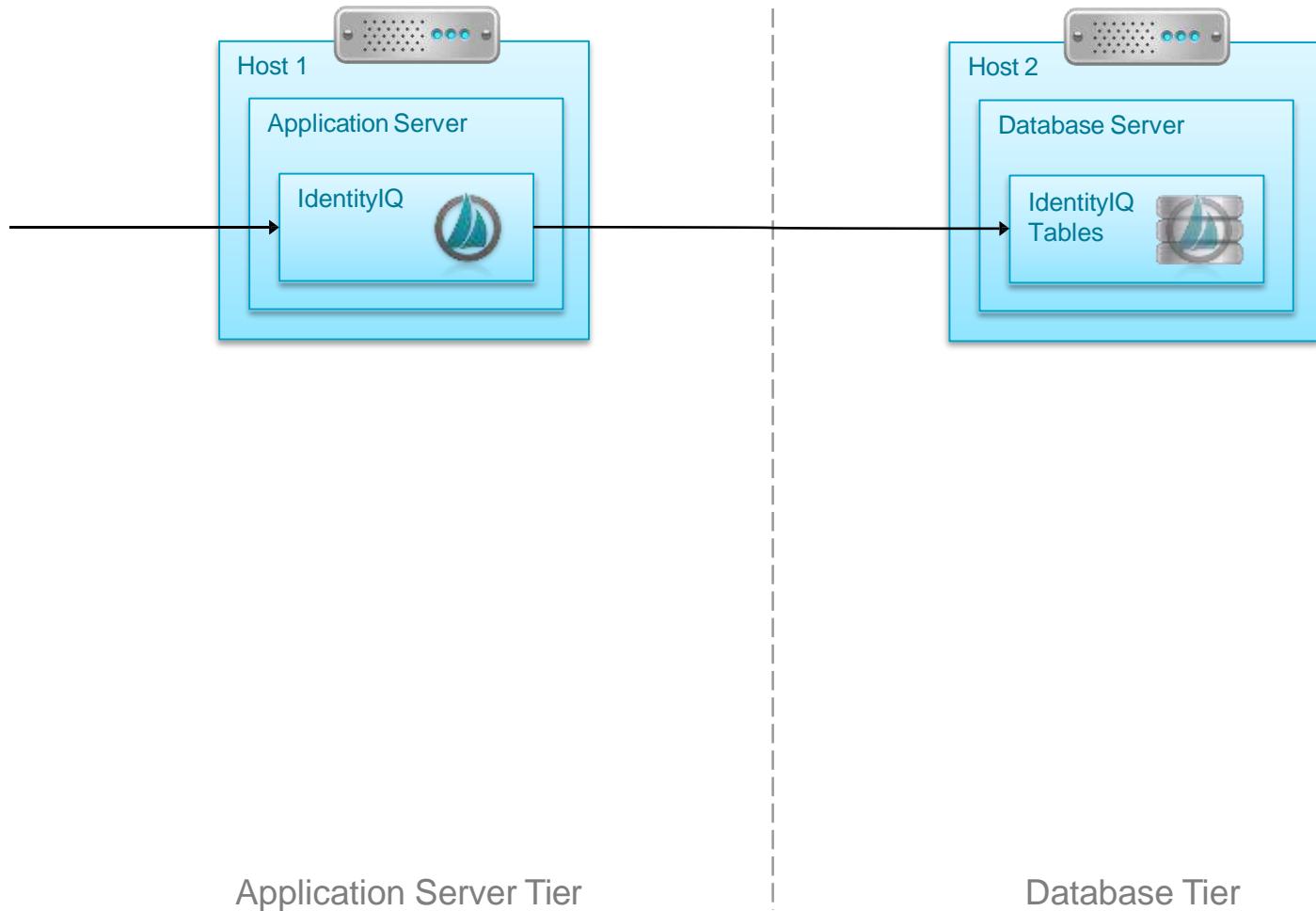
Architecture

Simplest Model



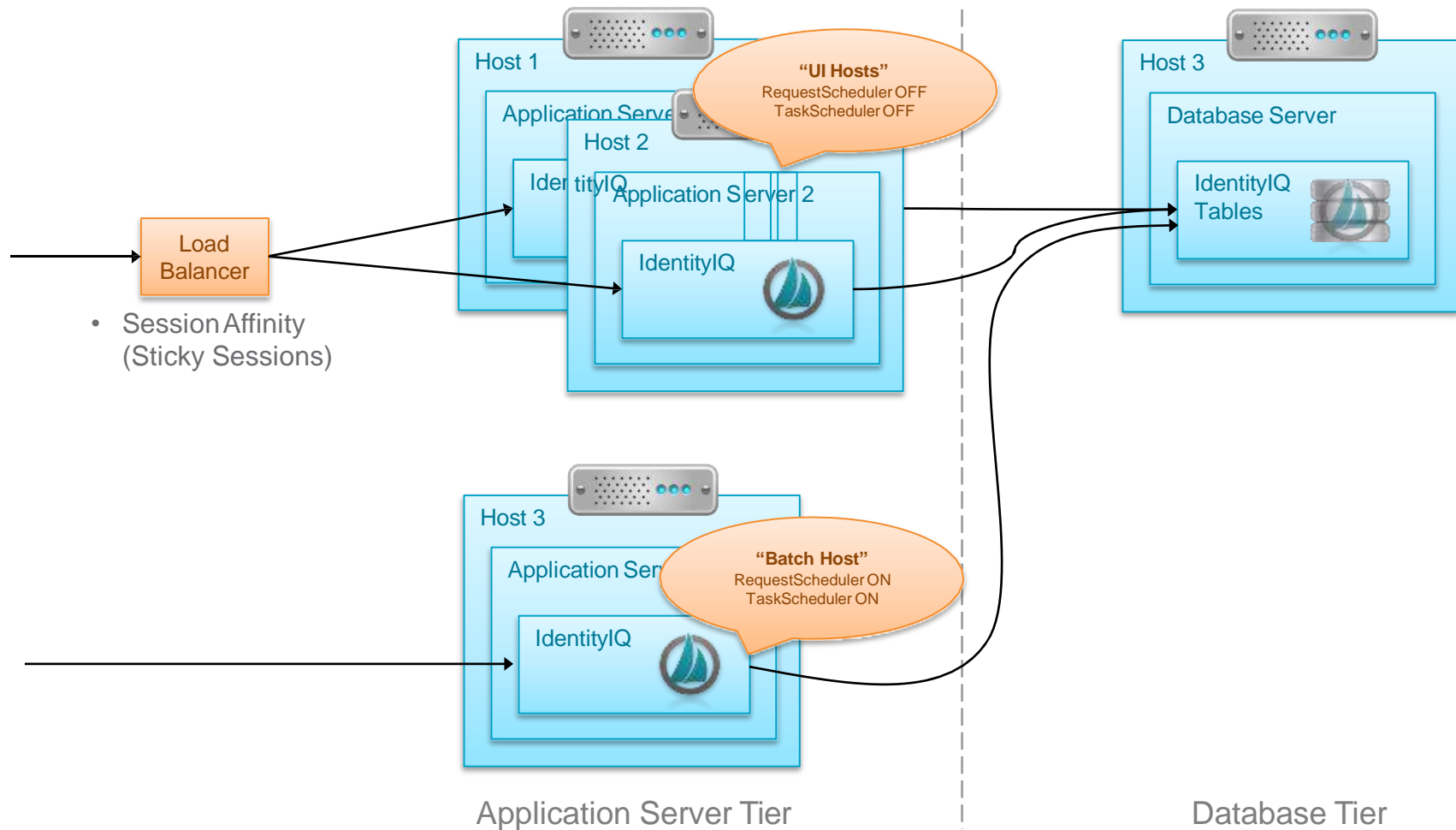
Architecture

Processing and Storage Segregation



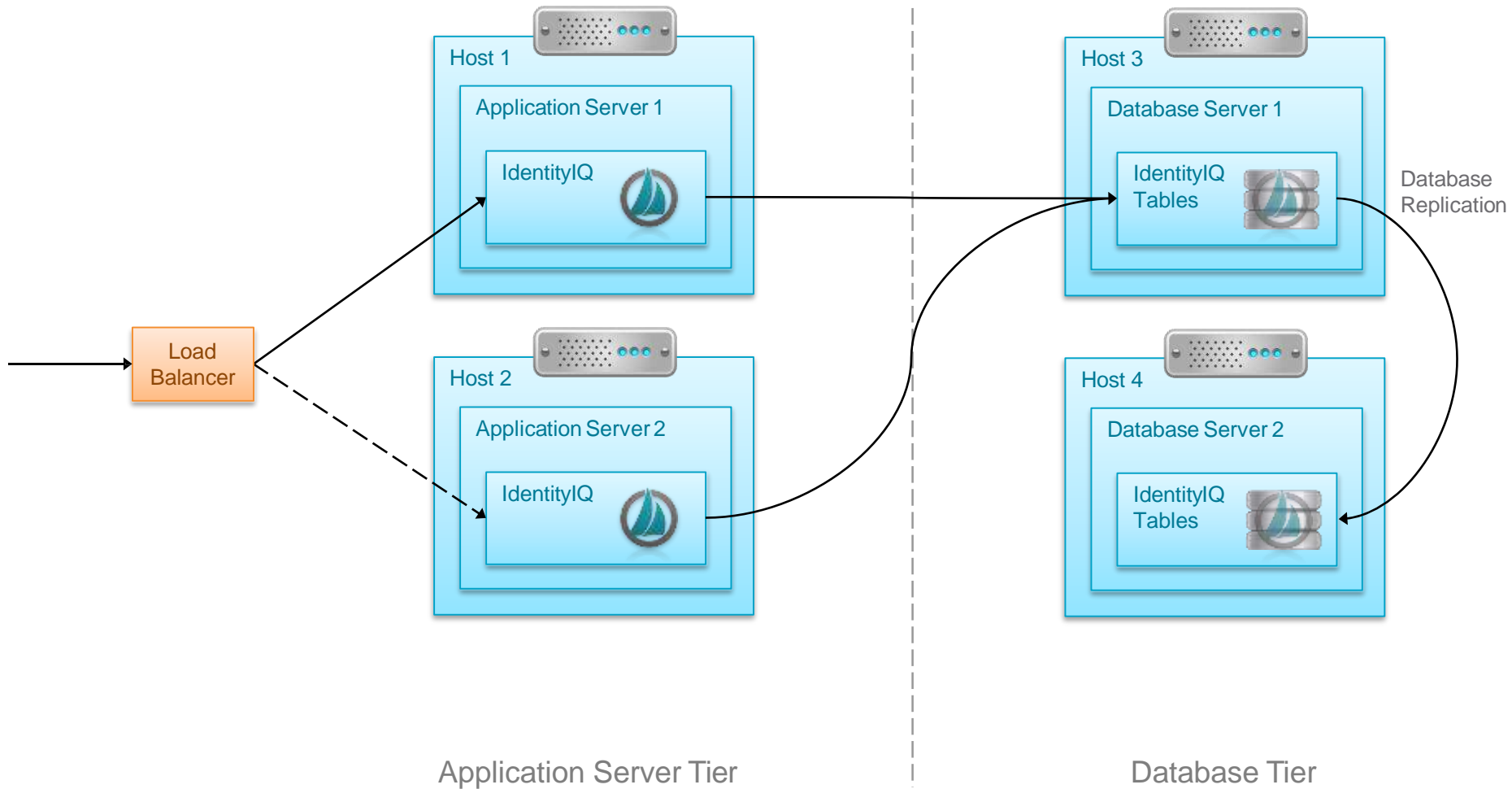
Architecture

Application Server Availability / Redundancy



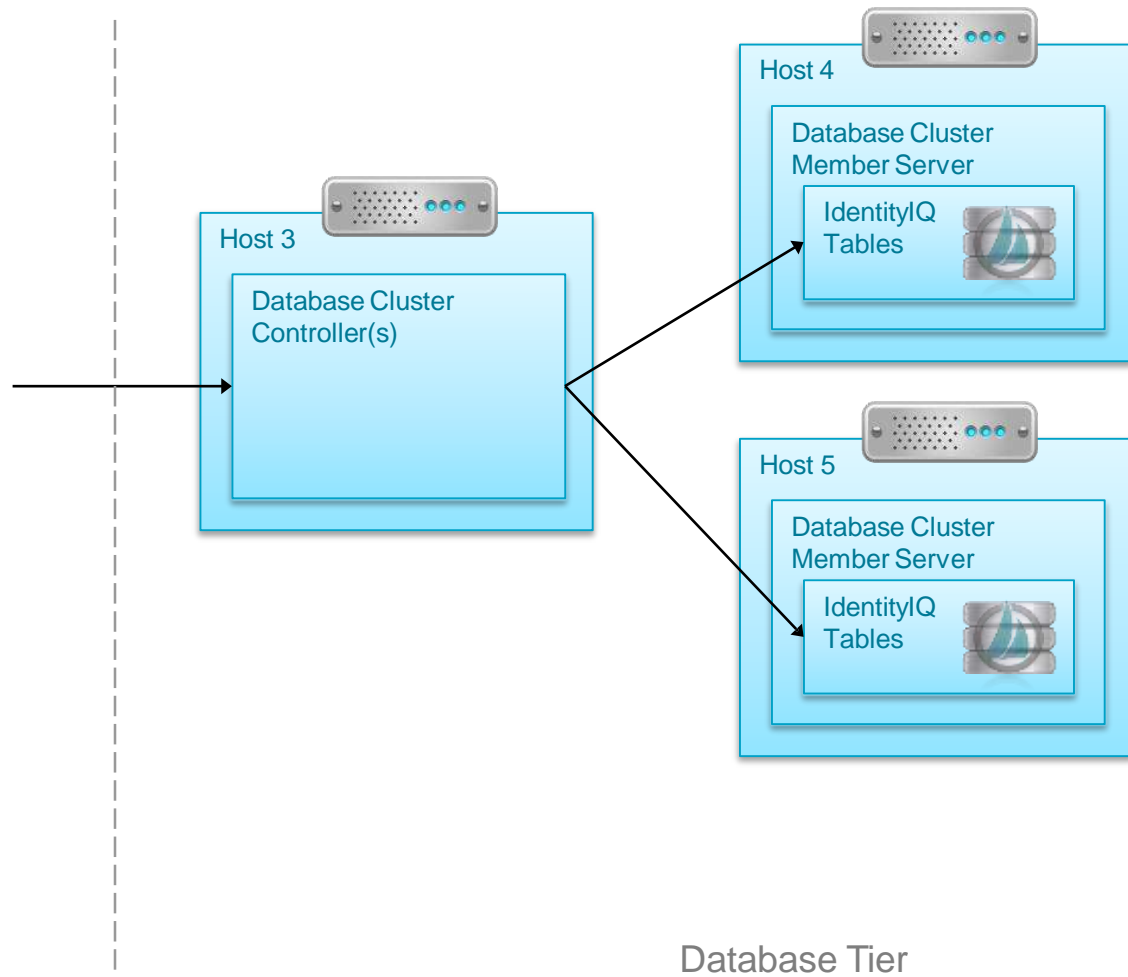
Architecture

Two-Tier Redundancy



Other Architectures

Database Clustering



Designating Batch/Task Hosts

IdentityIQ

- Controlled in the *Task* and the *Request* ServiceDefinition objects
 - Default, *hosts=global*, tasks and requests can run on any server

```
<ServiceDefinition created="1388105905677" hosts="global"
id="ff80808143318eba0143318f360d00f7" name="Task">
  <Description>
    Service definition for the Request processor service.
  </Description>
</ServiceDefinition>
```

- Specify batch hosts in both objects separated by commas

```
<ServiceDefinition created="1388105905701" hosts="HostA,HostB"
id="ff80808143318eba0143318f362500f8" name="Request">
```

```
<ServiceDefinition created="1388105905677" hosts="HostA,HostB"
id="ff80808143318eba0143318f360d00f7" name="Task">
```


Designating Batch/Task Hosts

Pre IdentityIQ

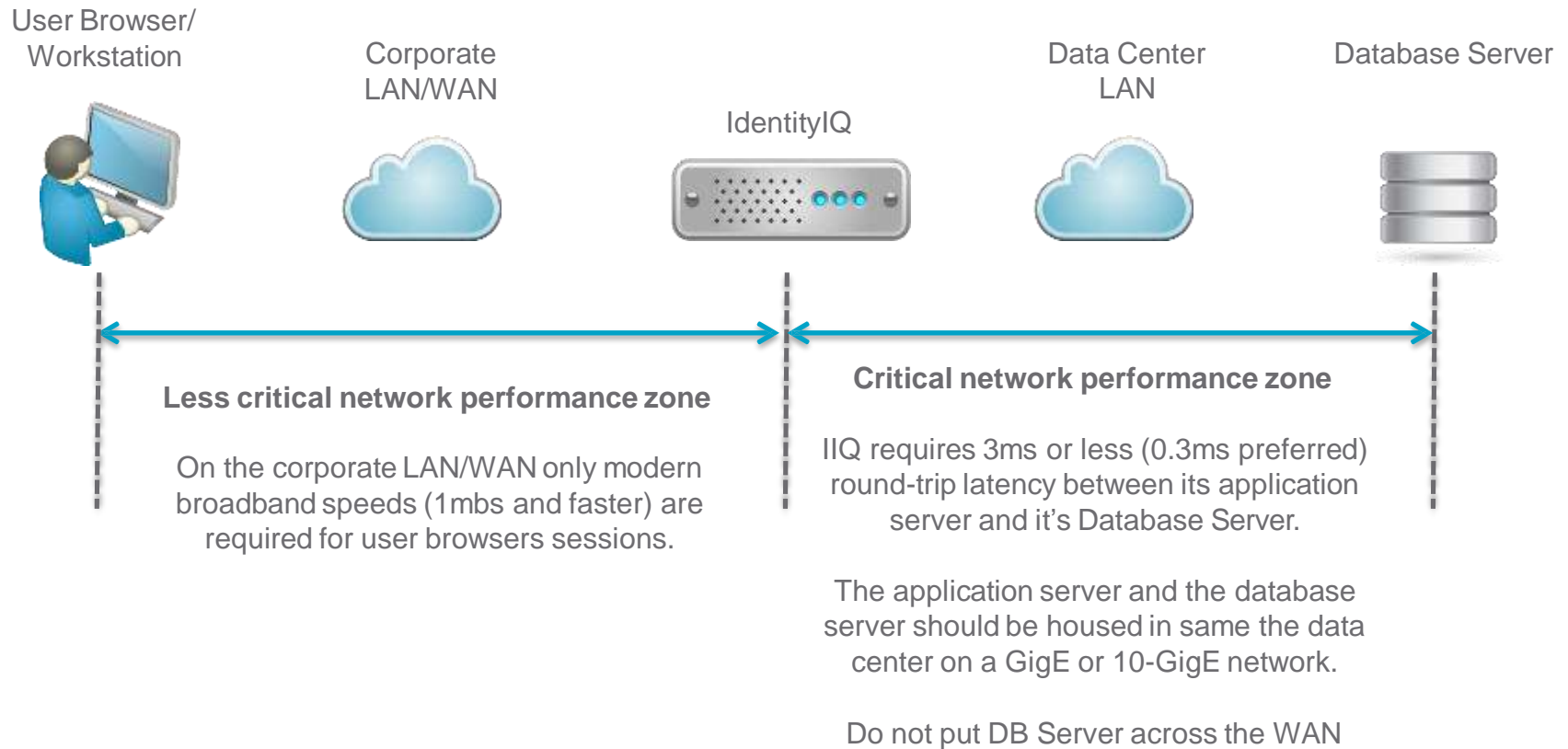
- Controlled in the properties file
identityiq/WEB-INF/classes/iiq.properties
- Add the properties
environment.taskSchedulerHosts
environment.requestSchedulerHosts
- Specify designated batch hosts (non UI hosts) in both lists, separated by commas

environment.taskSchedulerHosts=**HostA,HostB**
environment.requestSchedulerHosts=**HostA,HostB**

Note: The preferred method for designating batch/task hosts is through the ServiceDefinition objects. Specification via properties file is subject to future deprecation.

Deployment Database Considerations

- Network Proximity (latency) to the Database Server is extremely important for IdentityIQ



Questions?

Course Materials and Installation

Review

- Downloads

- Slide PDFs
- Exercise PDFs
- Fundamentals of IdentityIQ Implementation Virtual Machine
- VM Player (optional)

- Installation

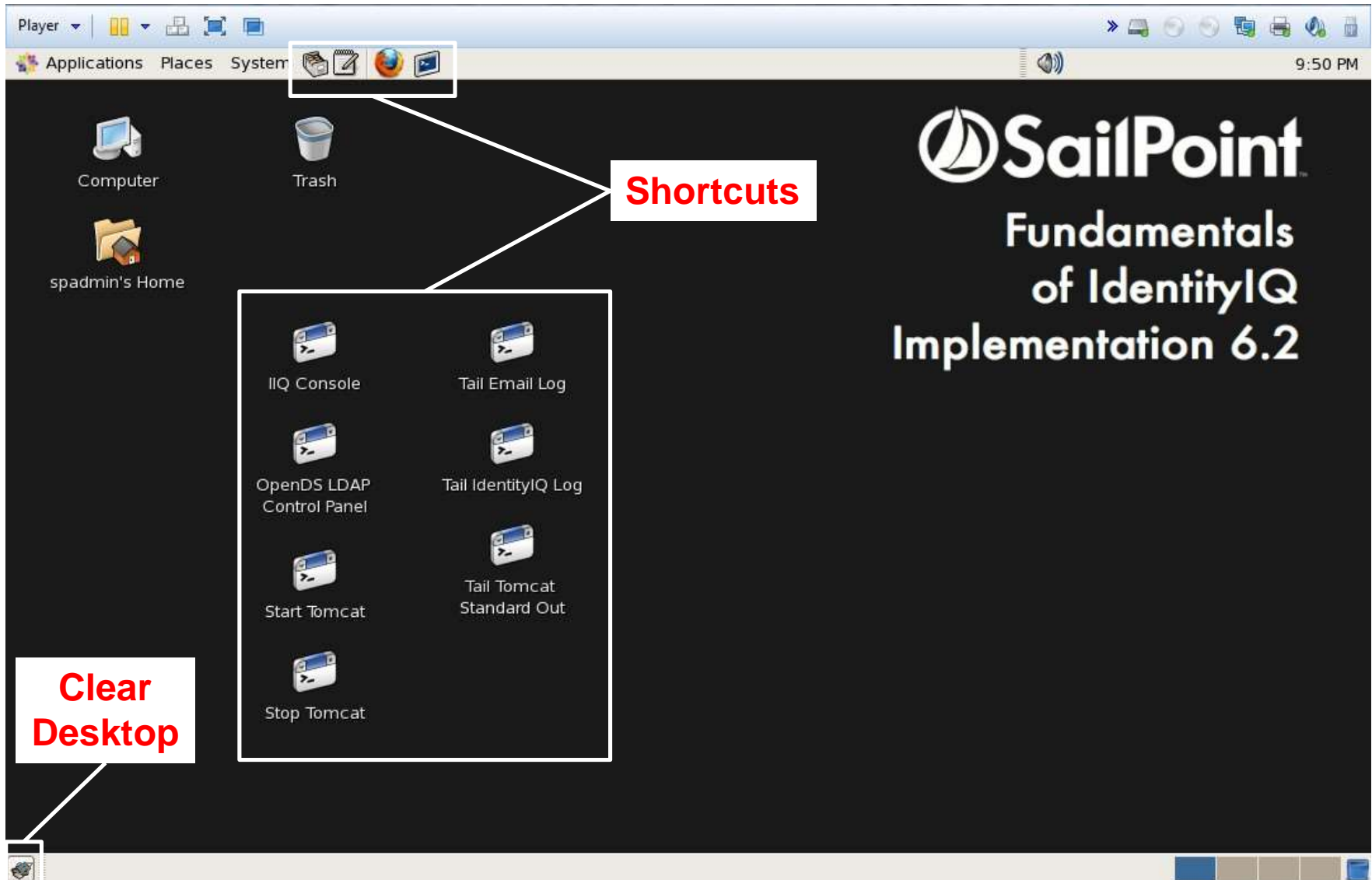
- Copy VM ZIP File to your machine
- Unzip
- Launch VM

- Linux syntax help

- Exercise book, Appendix: *Basic Linux Commands*

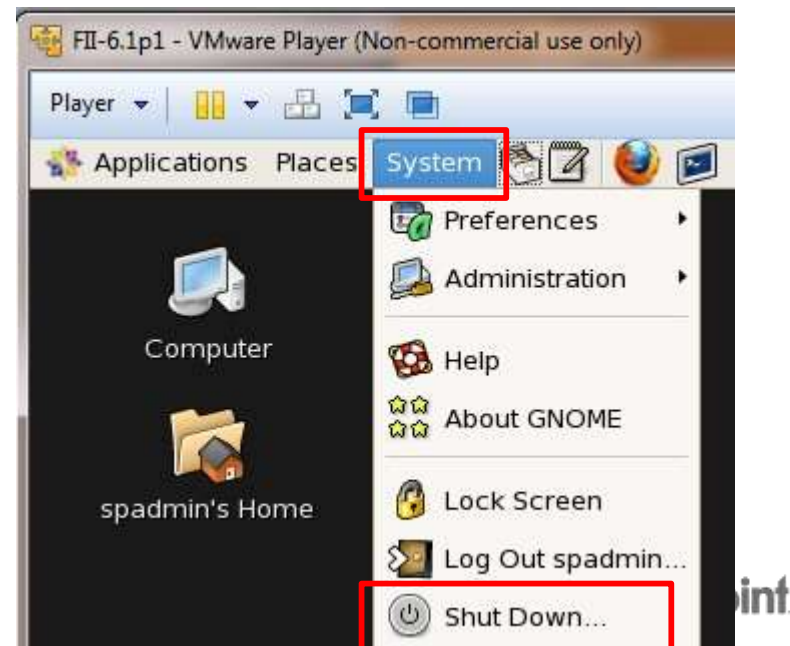
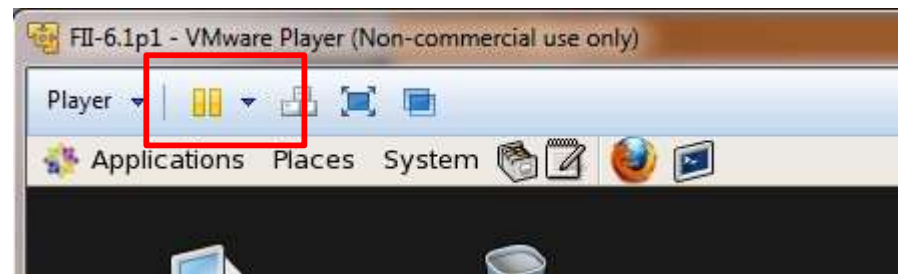
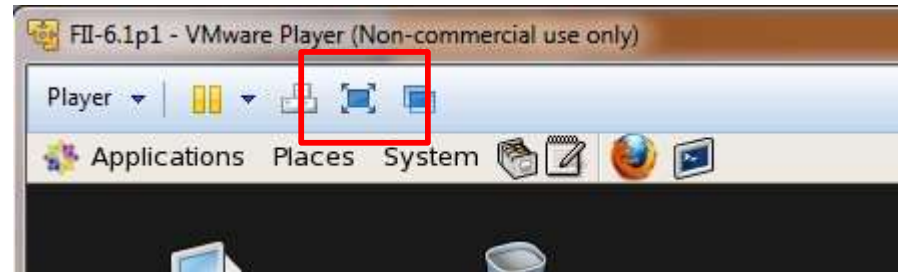
Course Environment

Your Virtual Machine



VM Care and Feeding

- For more screen space, expand Player
- To close VM
 - During class
 - Suspend through Player
 - ...or leave it open
 - Upon class completion (or for VM problems)
 - Shut down/Restart through Linux



Exercise Preview

Section 1, Exercises 1, 2, & 3

- **Exercise 1: Installing IdentityIQ**
 - Install IdentityIQ war file
 - Configure the database
 - Initialize and verify IdentityIQ
- **Exercise 2: Patching IdentityIQ**
 - This class runs on GA. We are not patching. Use this section as reference for future patching.
- **Exercise 3: Configuring IdentityIQ**
 - Redirect email
 - Configure auditing
 - Configuring logging