Product Architecture, Installation, and Deployment

Fundamentals of IdentityIQ Implementation

IdentityIQ 7.0

SailPoint™

Overview

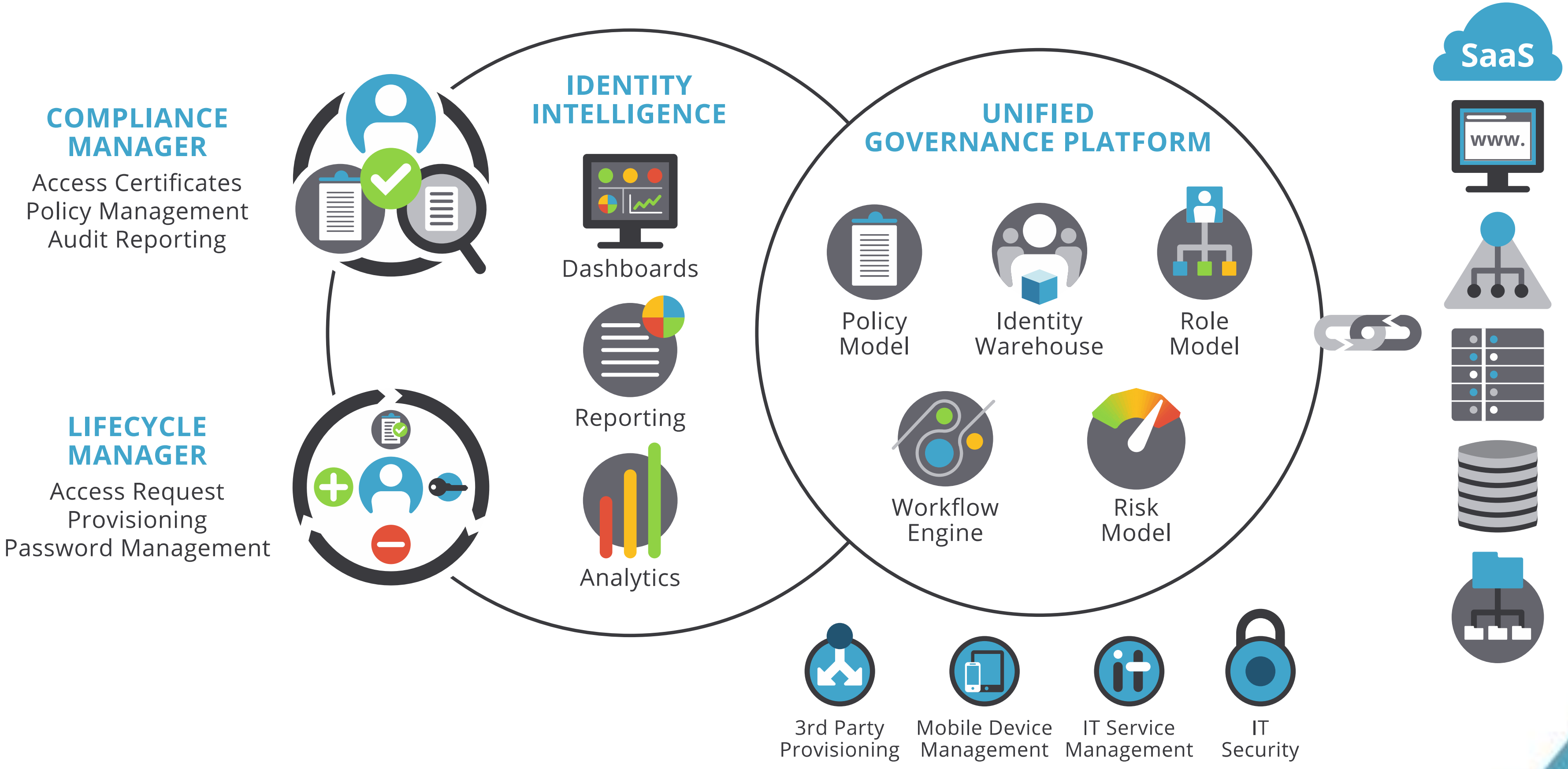
Product Architecture, Installation, and Deployment

- Product architecture overview
- Deployment strategy and environment management
- Deployment characteristics of IdentityIQ
  - Task Hosts and Request Hosts
  - Deployment Consideration for Database
  - High End Deployments (Redundancy)
- Installation
  - Includes considerations when adding business specific attributes

ARCHITECTURE

IdentityIQ Product Components

Review



COMPLIANCE MANAGER

Access Certificates
Policy Management
Audit Reporting

LIFECYCLE MANAGER

Access Request
Provisioning
Password Management

IDENTITY INTELLIGENCE

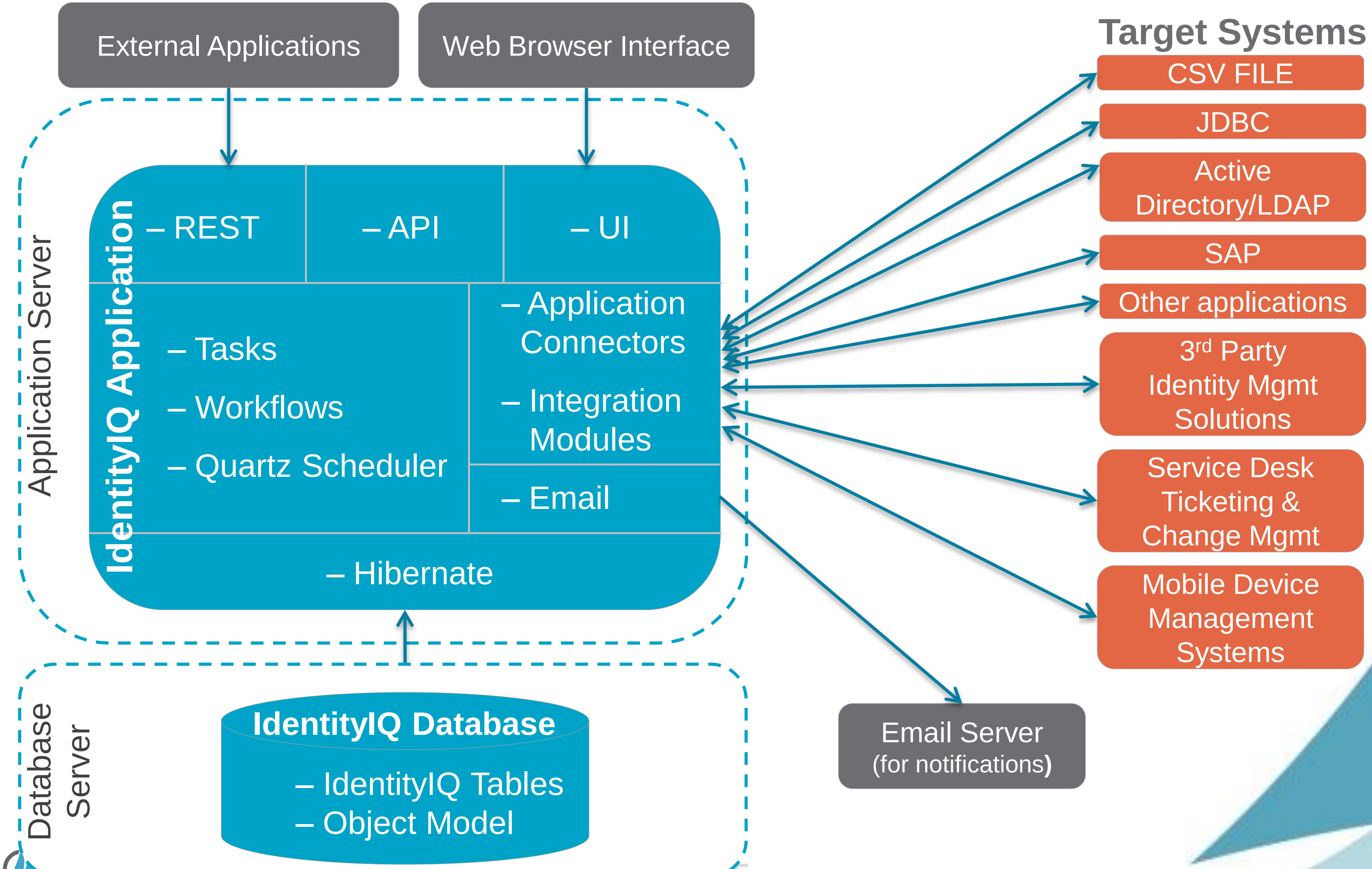
Dashboards
Reporting
Analytics

UNIFIED GOVERNANCE PLATFORM

Policy Model
Identity Warehouse
Role Model
Workflow Engine
Risk Model

3rd Party Provisioning
Mobile Device Management
IT Service Management
IT Security

Detailed Architecture Overview



External Applications

Web Browser Interface

Application Server

IdentityIQ Application

– REST

– API

– UI

– Tasks

– Workflows

– Quartz Scheduler

– Application Connectors

– Integration Modules

– Email

– Hibernate

Database Server

IdentityIQ Database

– IdentityIQ Tables

– Object Model

Target Systems

- CSV FILE
- JDBC
- Active Directory/LDAP
- SAP
- Other applications
- 3rd Party Identity Mgmt Solutions
- Service Desk Ticketing & Change Mgmt
- Mobile Device Management Systems

Email Server
(for notifications)

Installation Components

- Java Runtime
- Application Server
- IdentityIQ Software running inside the Application Server
- Database Server

Sample IdentityIQ installation location (from the training VM):

/home/spadmin/tomcat/webapps/identityiq

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
  - IOS
  - Android
  - Windows Phone
  - Native Browser Blackberry

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

Extension Levels

Extension Target	Method	Knowledge Needed
IdentityIQ Objects	Configuration • Applications • Identity attributes • Rules • Etcetera	IdentityIQ Java XML
Web Application Objects	XHTML CSS Images	XHTML & JSF Web Design
Java	Compiled Code • Custom tasks • Custom connectors • Workflow libraries • Connectivity	Java

DEPLOYMENT

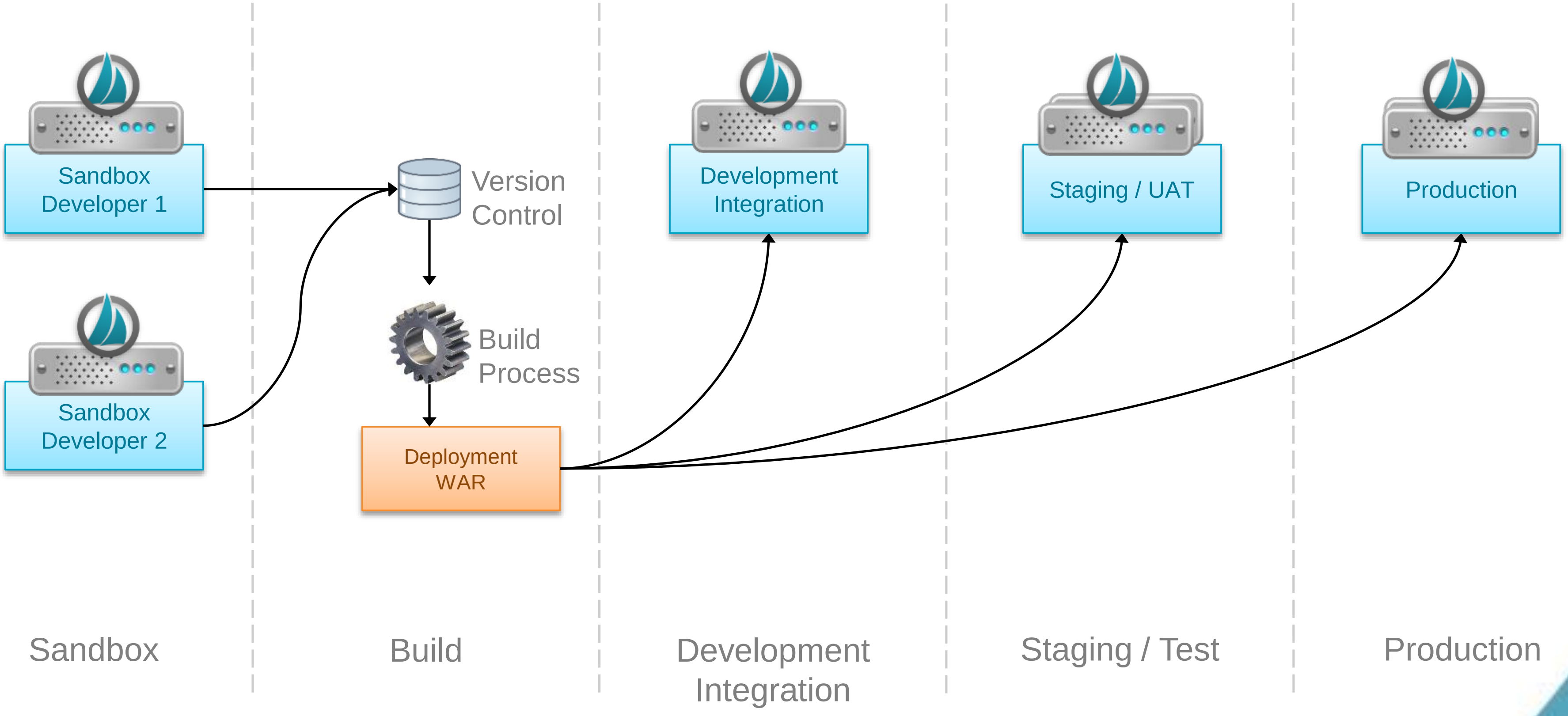
Deployment Strategy

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

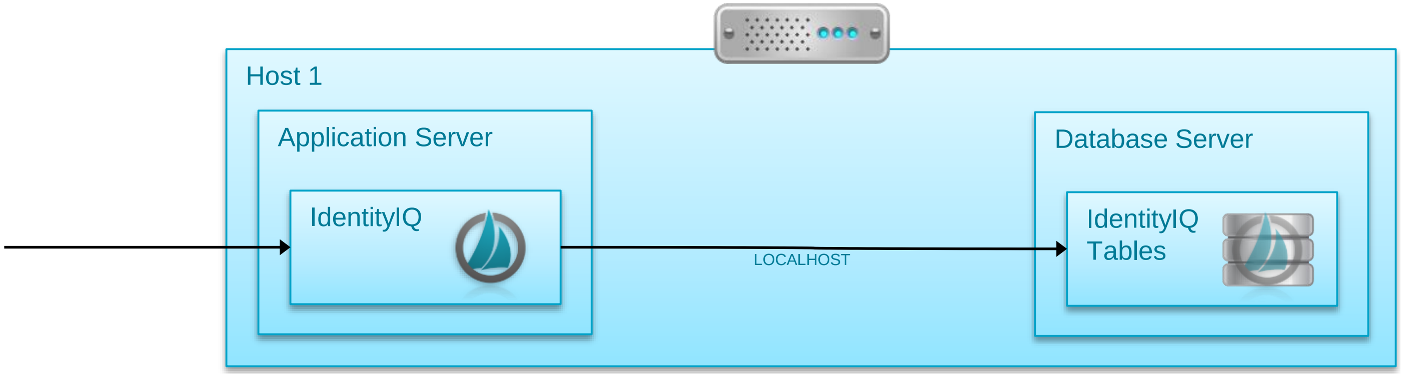
- SSB Process
  - Export objects from sandbox into XML files
  - Push XML files to version control system
  - Use the build tool to build *.war* from the version control directory
  - Release packaged war to additional environments

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

- Available on Compass

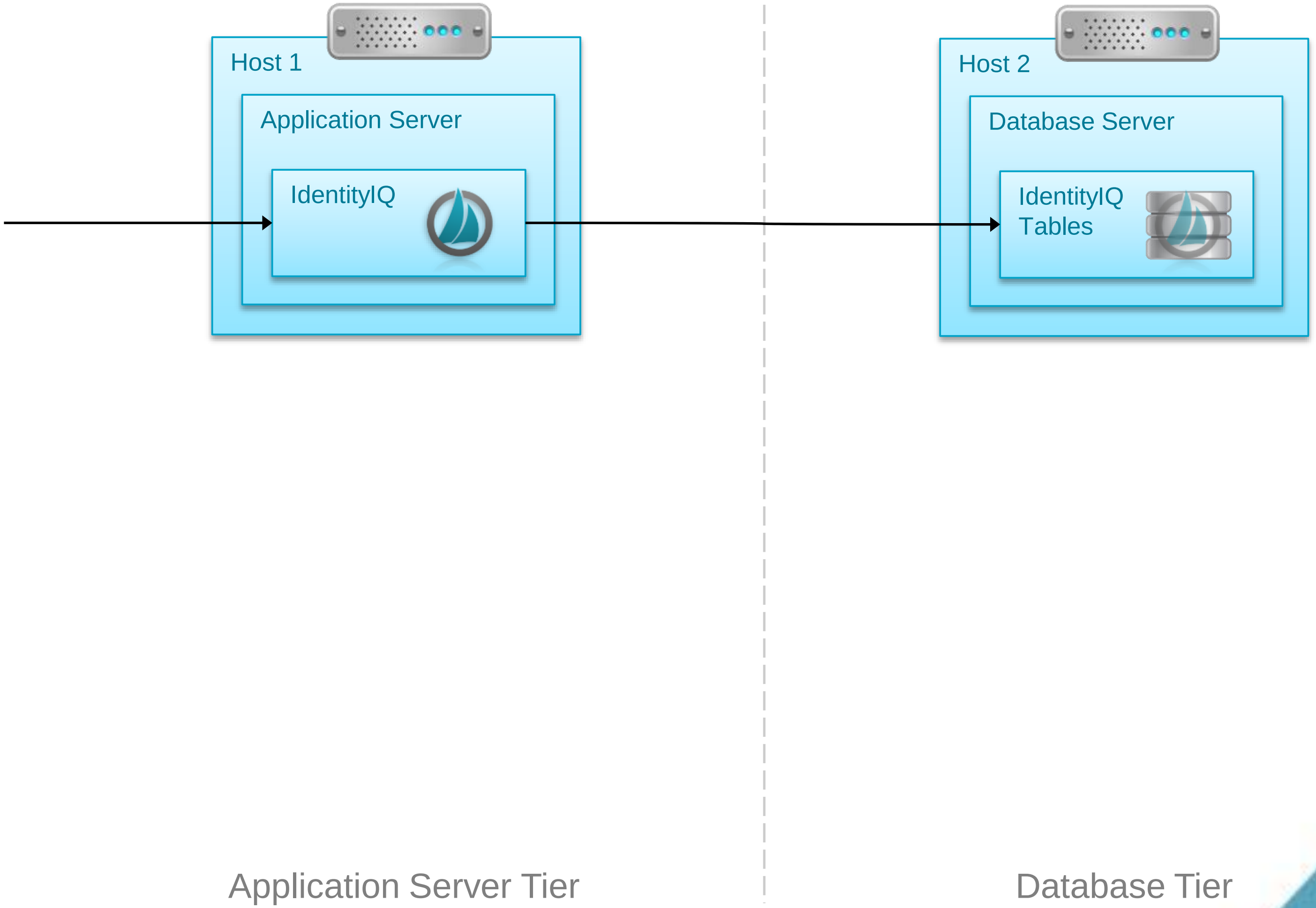
Architecture

Simplest Model



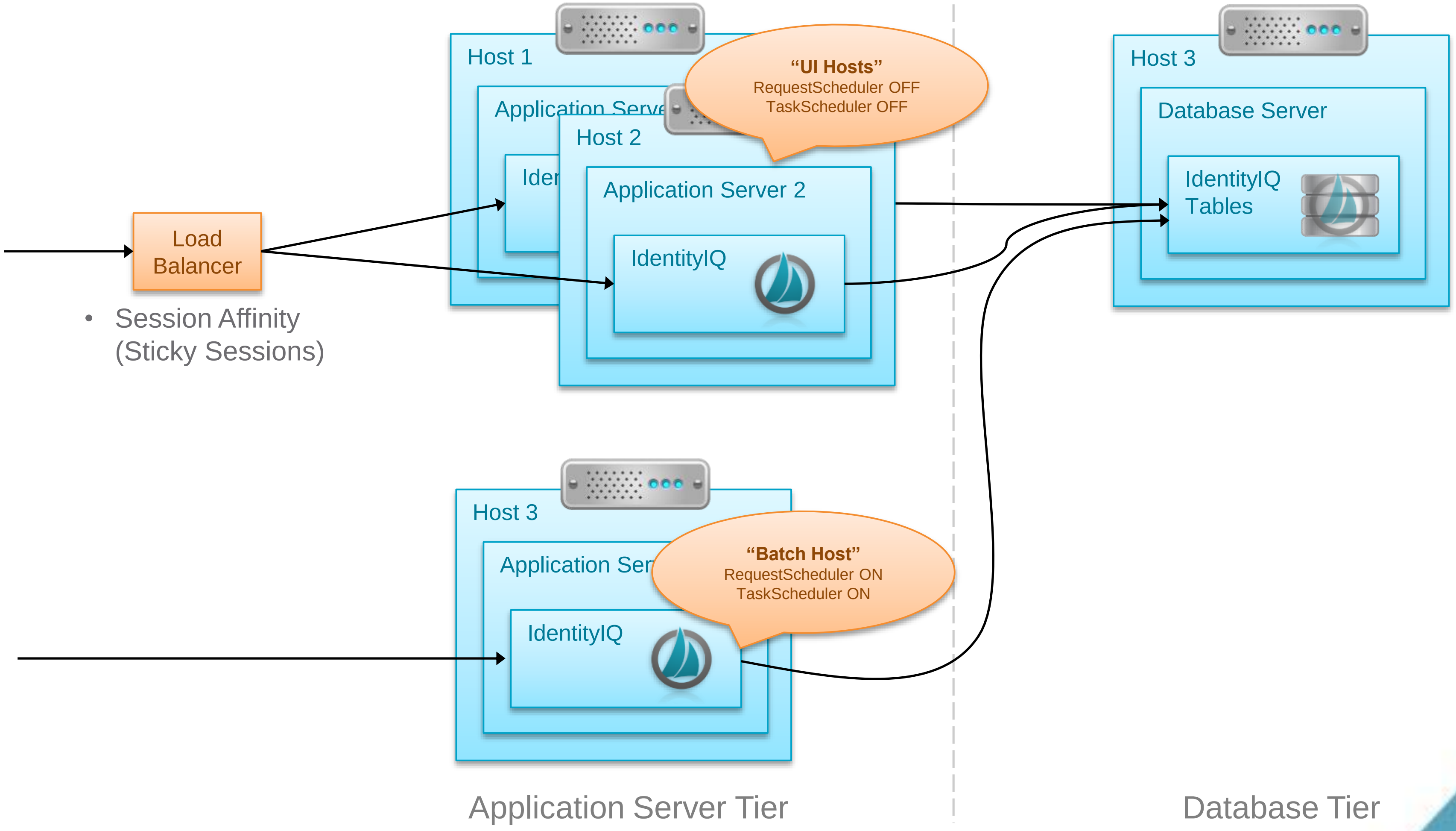
Architecture

Processing and Storage Segregation



Architecture

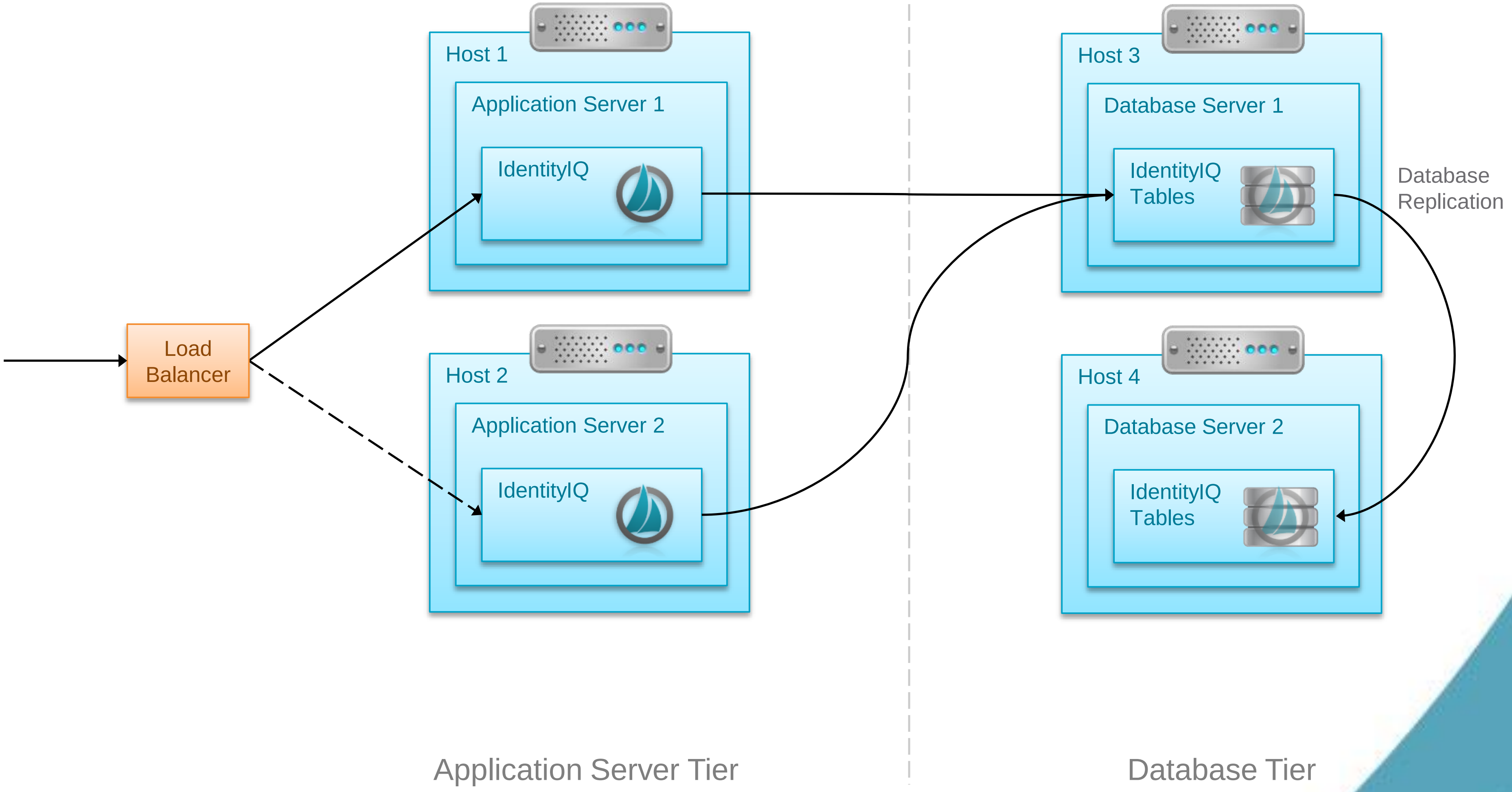
Application Server Availability / Redundancy



- Session Affinity (Sticky Sessions)

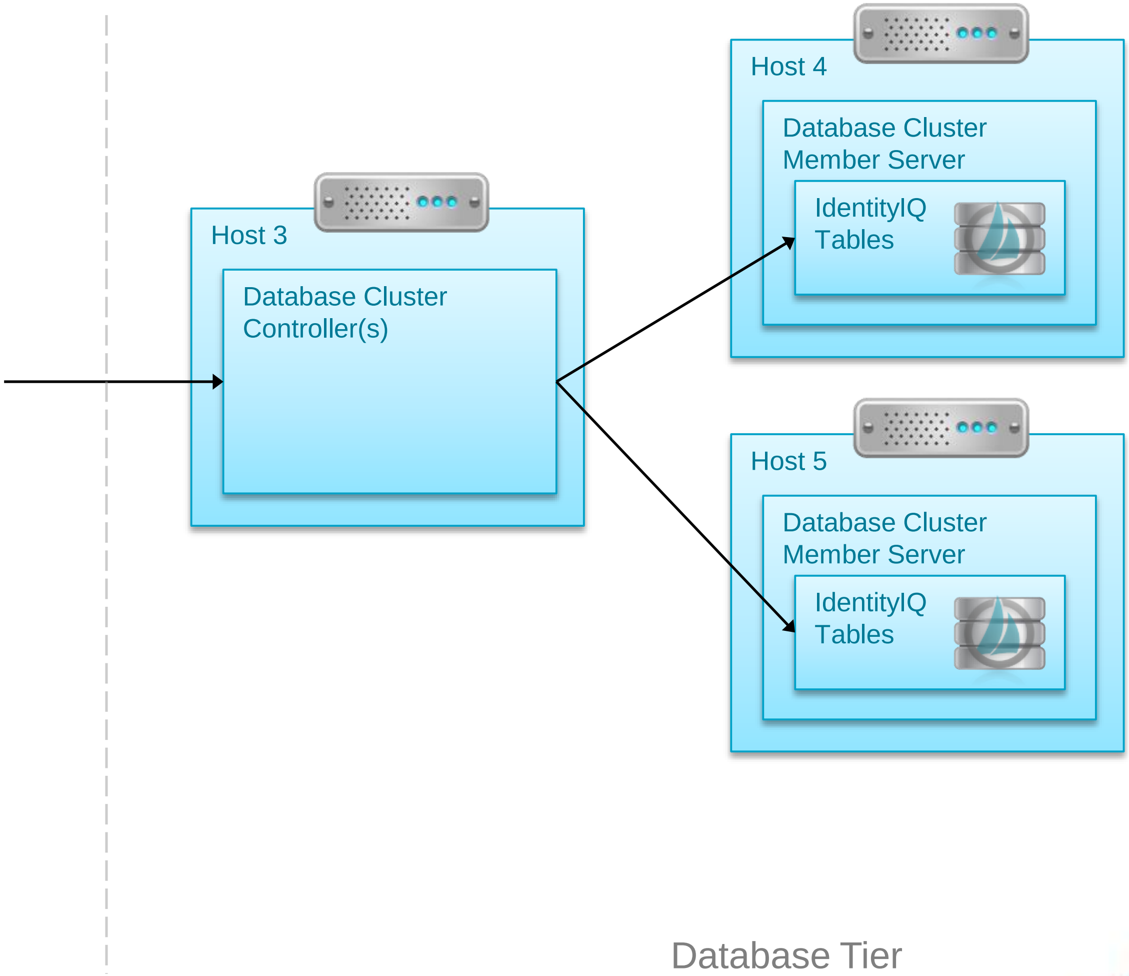
Architecture

Two-Tier Redundancy



Other Architectures

Database Clustering


Host 4
Database Cluster Member Server
IdentityIQ Tables
Host 3
Database Cluster Controller(s)
Host 5
Database Cluster Member Server
IdentityIQ Tables
Application Server Tier
Database Tier

Multi-host Deployments

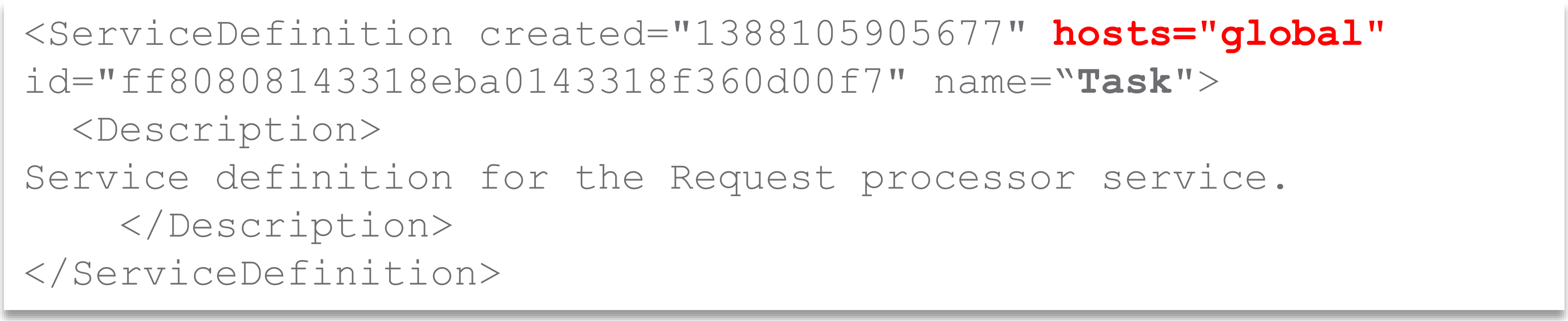
UI versus Batch Hosts

- Batch handles
  - Workflows
  - Tasks/reports
  - Certification generation
- UI hosts handles user interactions
  - Access Requests
  - Performing Certifications
  - Dynamic Analytics

Designating Batch/Task Hosts

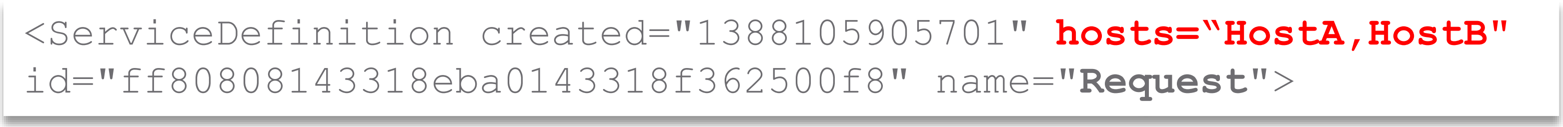
IdentityIQ 6.2+

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

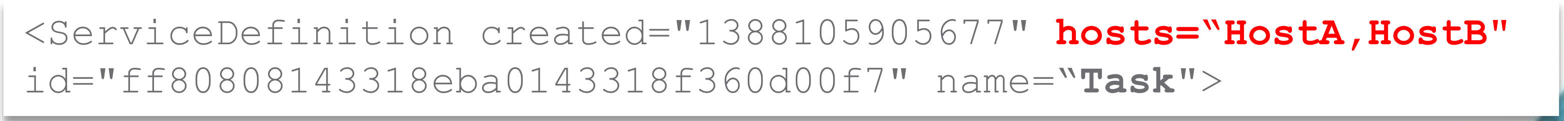


```
<ServiceDefinition created="1388105905677" hosts="global"
id="ff80808143318eba0143318f360d00f7" name=“Task">
  <Description>
Service definition for the Request processor service.
    </Description>
</ServiceDefinition>
```

- Specify batch hosts in both objects



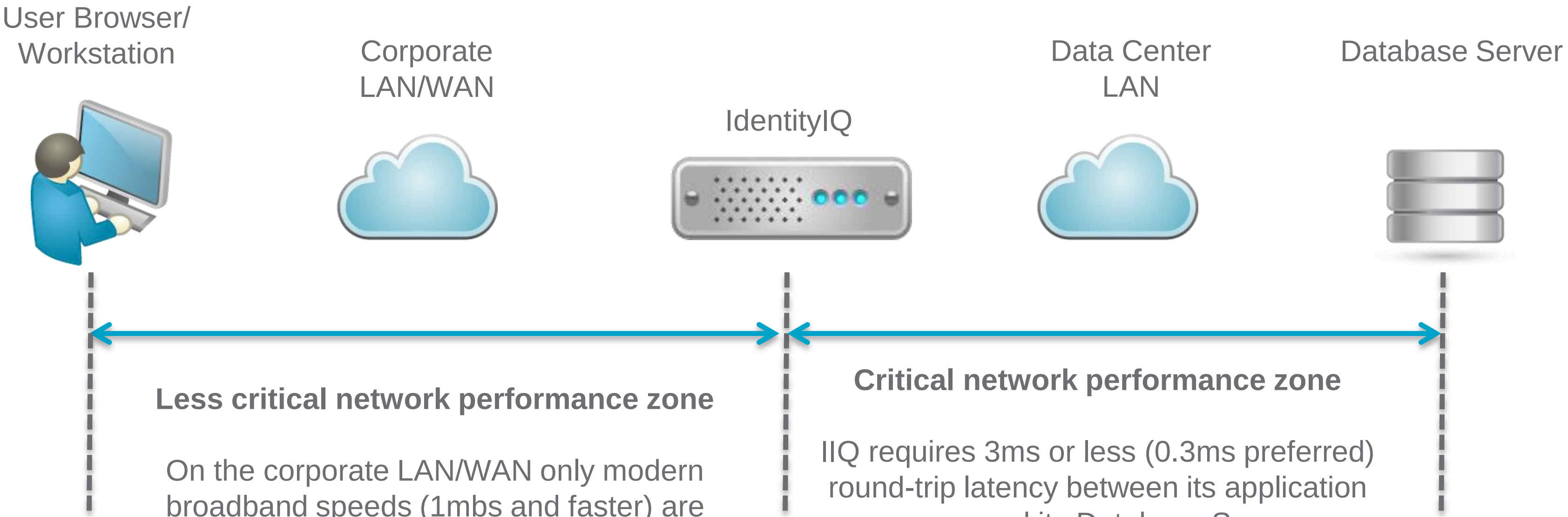
```
<ServiceDefinition created="1388105905701" hosts=“HostA,HostB"
id="ff80808143318eba0143318f362500f8" name="Request">
```



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

Deployment Database Considerations

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



Less critical network performance zone

On the corporate LAN/WAN only modern broadband speeds (1mbs and faster) are required for user browsers sessions.

Critical network performance zone

IIQ requires 3ms or less (0.3ms preferred) round-trip latency between its application server and its Database Server.

The application server and the database server should be housed in same the data center on a GigE or 10-GigE network.

Do not put DB Server across the WAN

INSTALLATION

IdentityIQ Installation Process

Overview

- Initial & Patch Deployment
  1. Deploy WAR file
  2. Modify and generate schema for IdentityIQ DB
  3. Create IdentityIQ database
  4. Configure iiq.properties
  5. Initialize default system objects
  6. Apply patches
- Ongoing Deployment & Operation
  - Initialize customized system objects
  - Deploy custom code
  - Deploy customized file-system artifacts

WAR File Deployment

- WAR (Web Application Archive)

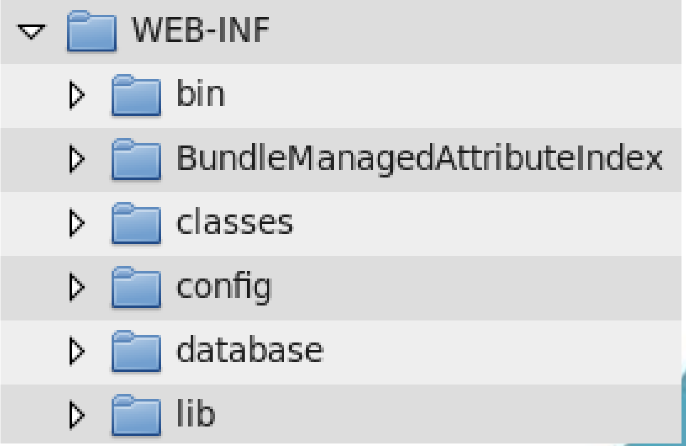
  File provided in product ZIP file

- 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 – identityiq/docs directory
    - PDFs of product docs
    - Java Doc – for developers
    - Online Help

WAR File Contents

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
    - IdentityIQ database schema files
  - \WEB-INF\config
    - Files used to bootstrap IdentityIQ
    - Example Files

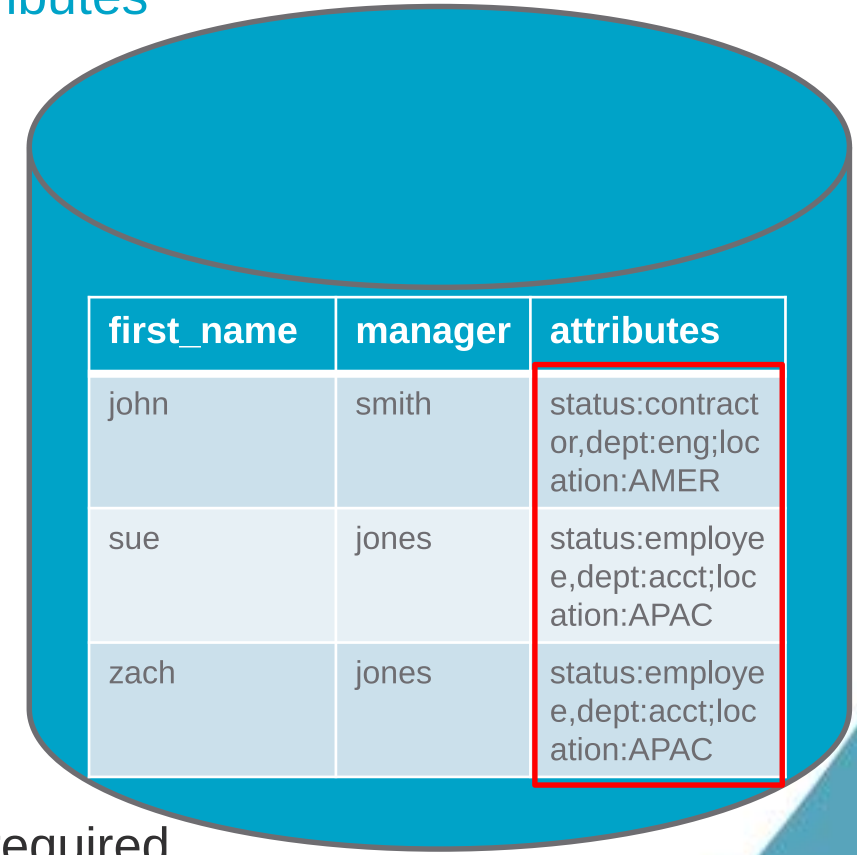


```
▽ WEB-INF
    ▷ bin
    ▷ BundleManagedAttributeIndex
    ▷ classes
    ▷ config
    ▷ database
    ▷ lib
```

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*

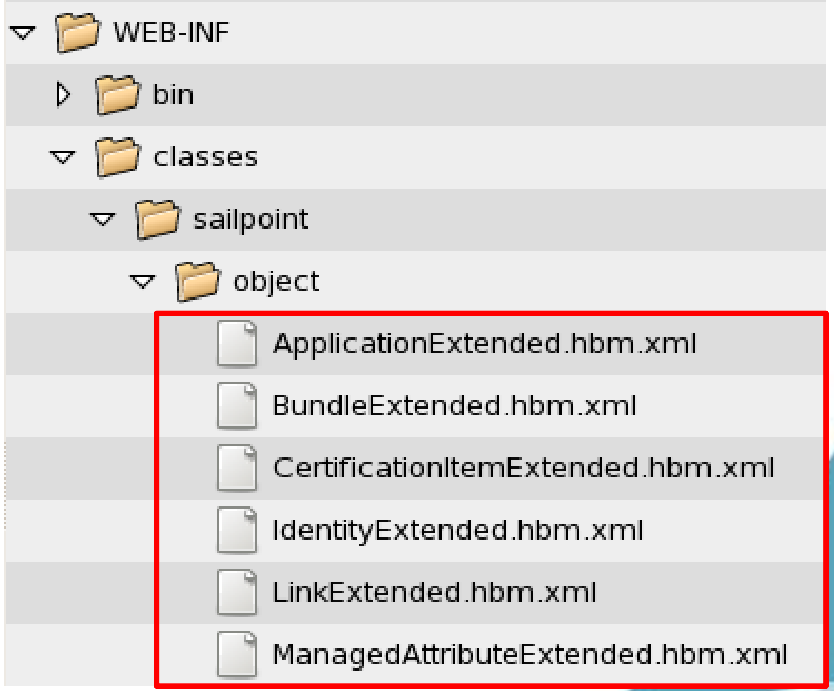


first_name	manager	attributes
john	smith	status:contractor,dept:eng;location:AMER
sue	jones	status:employee,dept:acct;location:APAC
zach	jones	status:employee,dept:acct;location:APAC

Configure Database Schema

Configure Searchable Extended Attributes

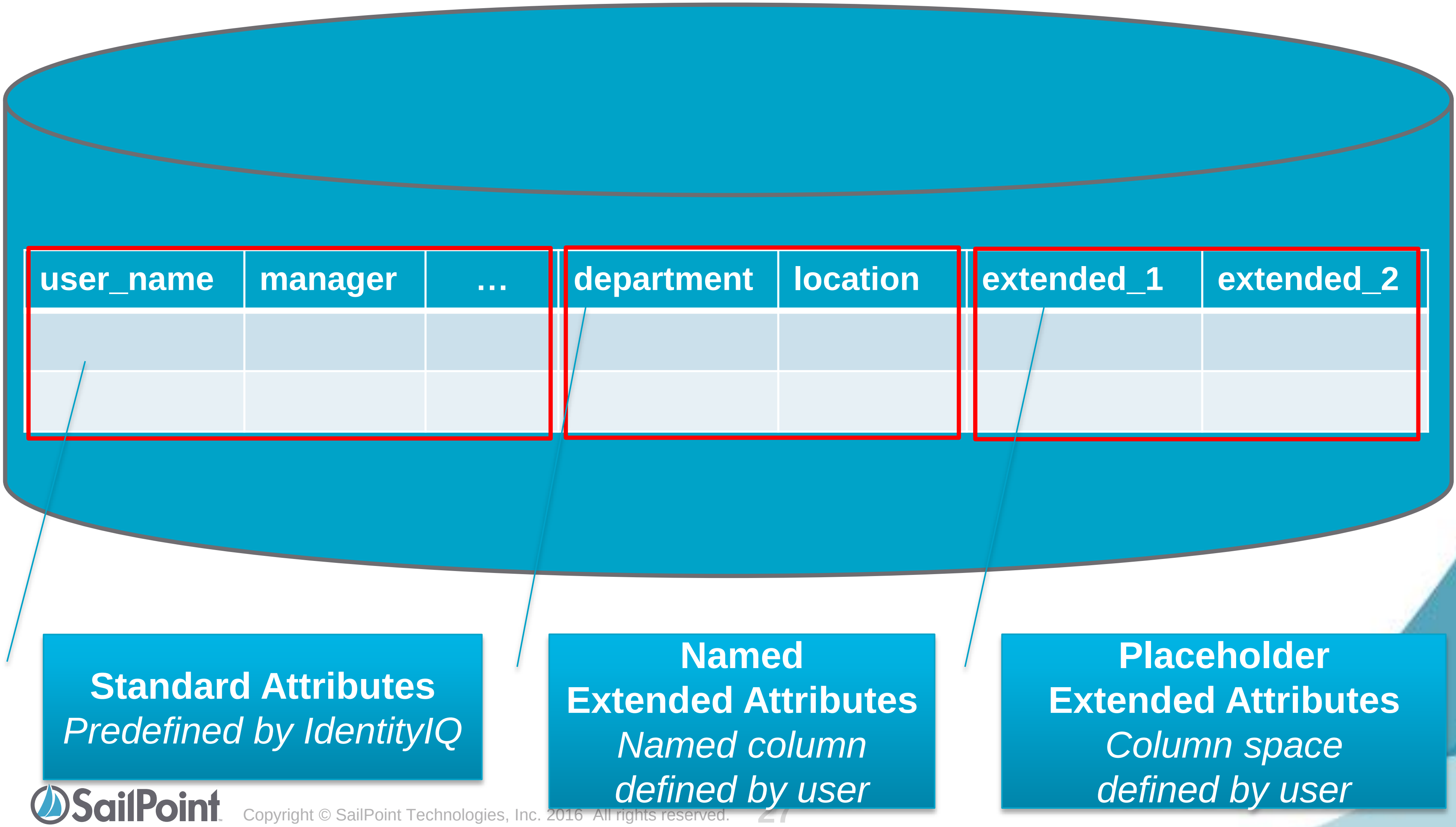
- Create columns for extended attributes in IdentityIQ database
  - Edit the appropriate Hibernate XML files
  - Generate schema
  - Generate database
- Add attributes to IdentityIQ and mark them as searchable



```
WEB-INF
  bin
  classes
    sailpoint
      object
        ApplicationExtended.hbm.xml
        BundleExtended.hbm.xml
        CertificationItemExtended.hbm.xml
        IdentityExtended.hbm.xml
        LinkExtended.hbm.xml
        ManagedAttributeExtended.hbm.xml
```

Database Schema Configuration

3 Types of Searchable Attributes



user_name	manager	…	department	location	extended_1	extended_2

Standard Attributes
Predefined 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, IdentityIQ matches to column
  - Object maximums: 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, IdentityIQ matches to next available placeholder column
  - Object maximums: 20 per object

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

Generate Database Schema

- Two schema options
  - Create schema (DDL) for a new IdentityIQ database
    - /WEB-INF/bin/iiq schema
  - Create schema (delta DDL) to update an IdentityIQ database
    - /WEB-INF/bin/iiq extendedSchema
- Console-based Schema Creation
  - Assures unique schema to each deployment
  - Input is Hibernate XML files
  - Generates DDL for all supported Databases
    - MySQL
    - Oracle
    - MS SQL Server
    - DB2
  - Filenames
    - create_identityiq_tables.<databasetype>
      example: create_identityiq_tables.mysql

Create IdentityIQ Database

- Create a database and all the necessary tables for IdentityIQ
  - Use your database tools of choice
  - Leverage database scripts
- Database Scripts
  - 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-7.0.mysql** **(default)**
      - **drop_identityiq_tables-7.0.mysql**
      - **upgrade_identityiq_tables.mysql**
      - **post_upgrade_identityiq_tables.mysql**

Note: If generating custom scripts, take care to load correct files. Look for correct name or date/time stamps to ensure you are using the most recently generated files.

Configure IdentityIQ Properties

Identify Database to IdentityIQ

/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 set up 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

Questions?

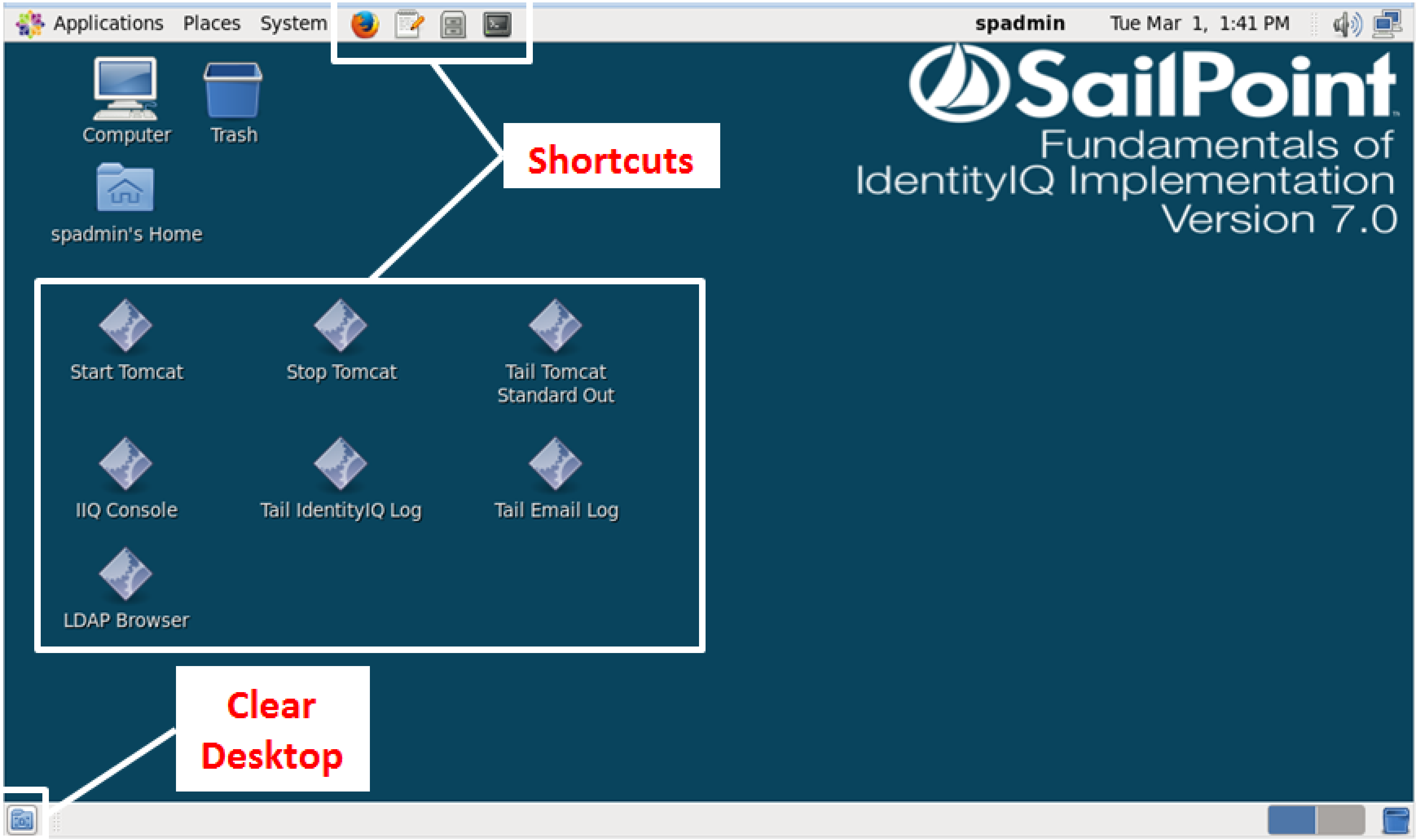
Course Materials and Installation

Review

- Downloads
  - Slide PDFs
  - Exercise PDFs
  - Fundamentals of IdentityIQ Implementation Virtual Machine
- 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

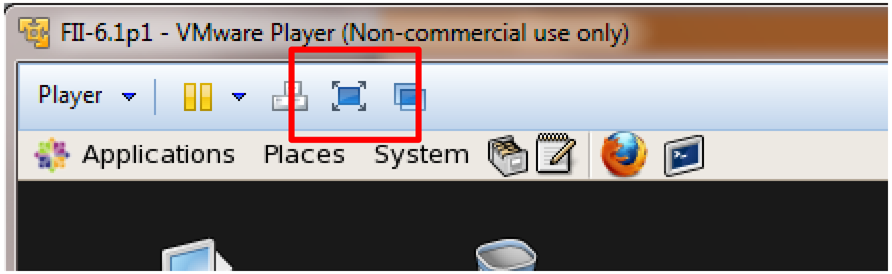


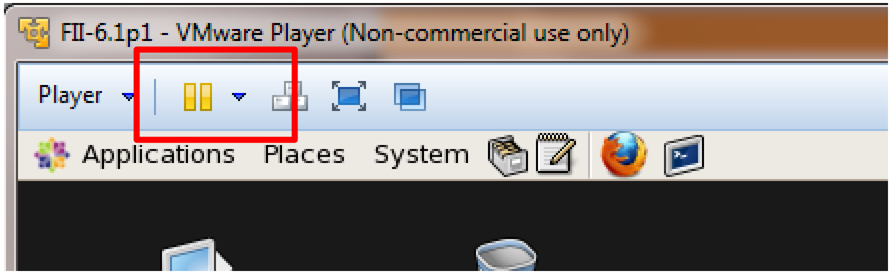
Shortcuts

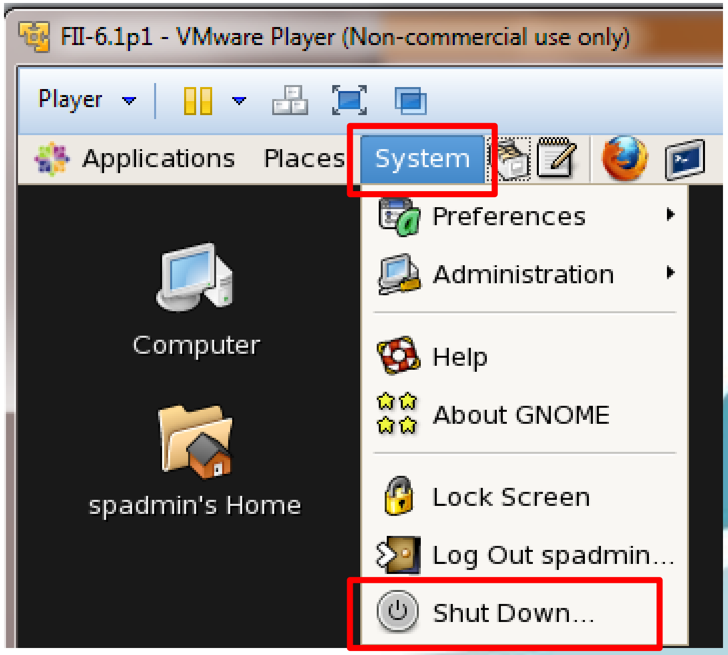
Clear Desktop

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
- Exercise 3: Configuring IdentityIQ
    - Redirect email
    - Configure auditing
    - Configuring logging