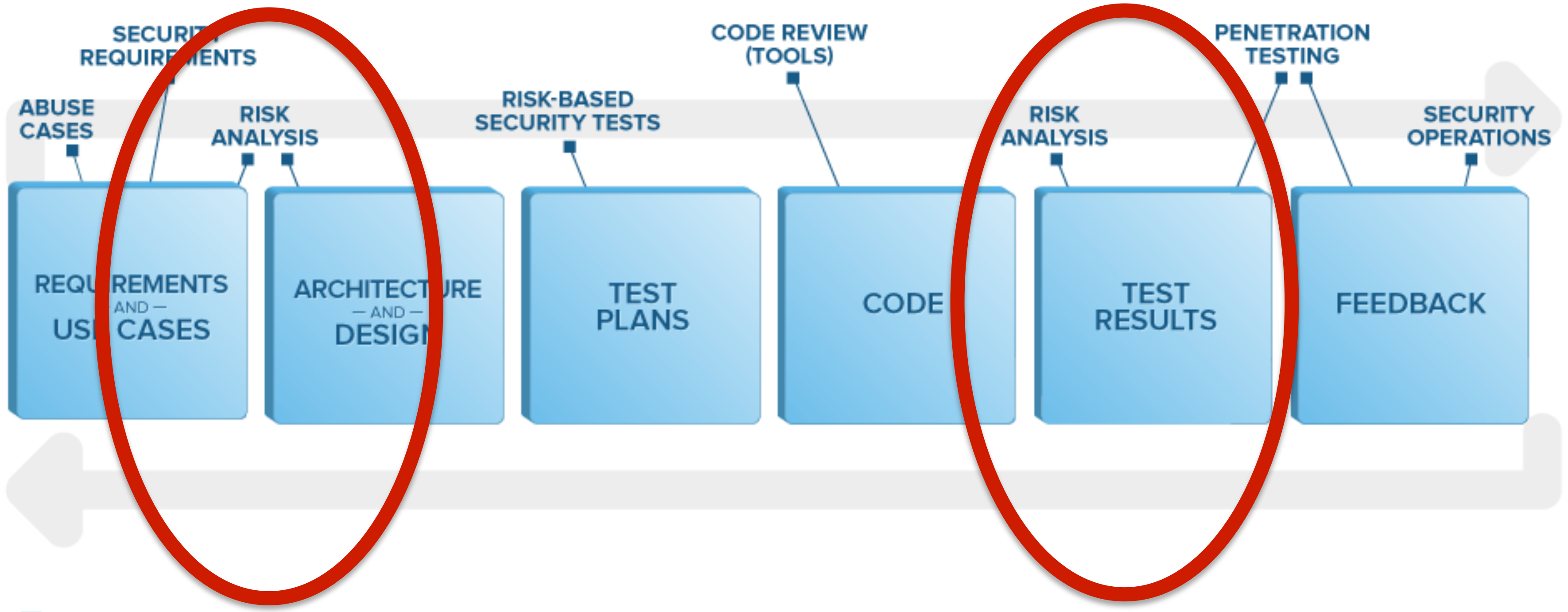
Threat Modeling

JIM DELGROSSO
PRINCIPAL CONSULTANT

@JIMDELGROSSO

What Is Threat Modeling?

A software design analysis capable of finding flaws



cigital

Threat Model Process

Threat Modeling Vocabulary

Asset

Security Control

Threat Agent

Attack Surface

Threat

Likelihood

Impact

Mitigation

Traceability Matrix

cigital

|

Threat Model Process

Define scope and depth of analysis

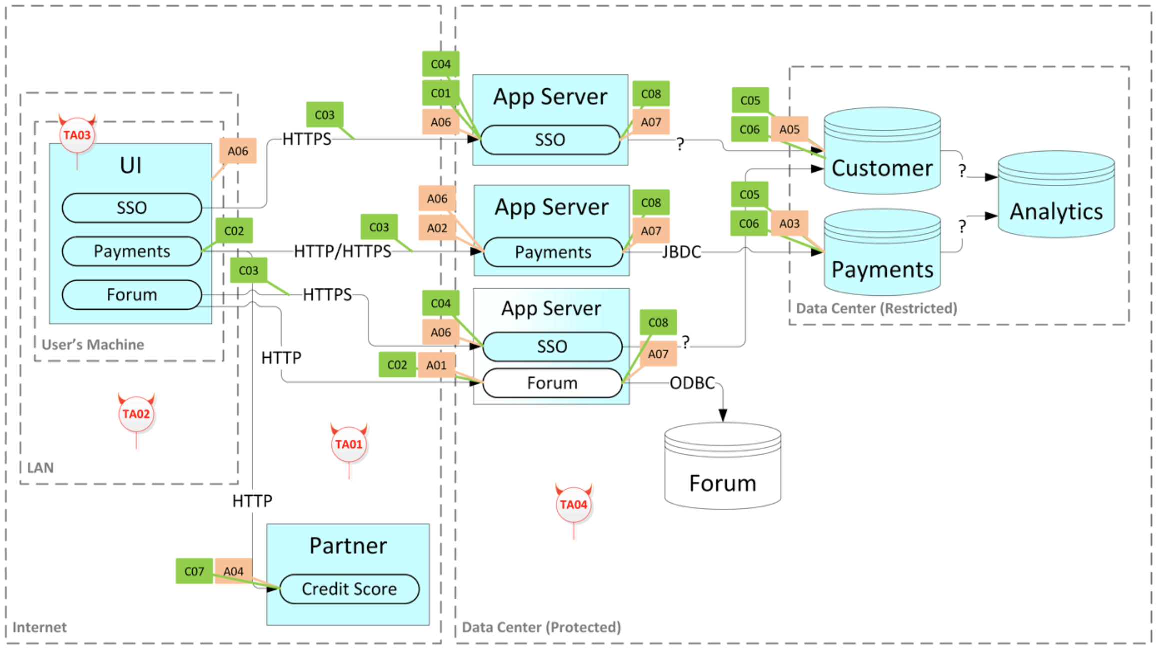
Gain understanding of what is being modeled

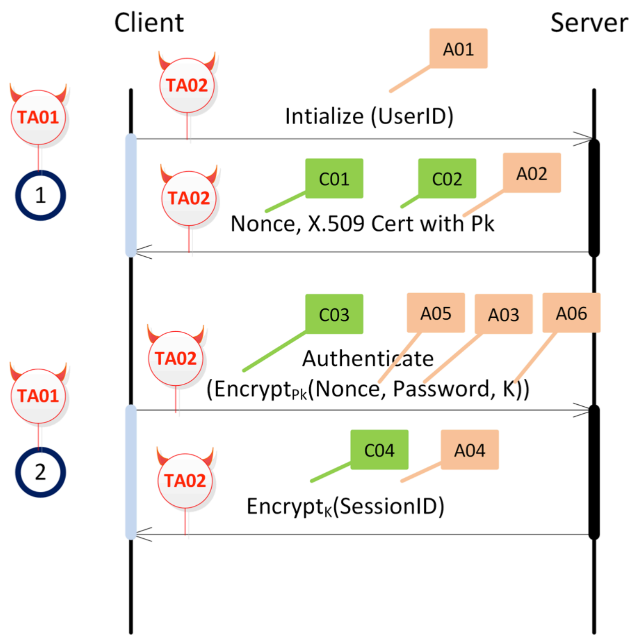
Model the threat structure

Interpret the threat model

Create the Traceability Matrix

Different Types Of Threat Models





System
Threat Model

Protocol/API
Threat Model

cigital

System Threat Models

Decompose And Model The System

- Understand how the system works (before trying to break it)
  - Who uses the system
  - What are the business goals/risks
  - What are the dependencies between systems
    - What other systems (components) does this system make use of
    - What other systems (components) use this system
- Review (some) development documentation
- Interview members of various teams

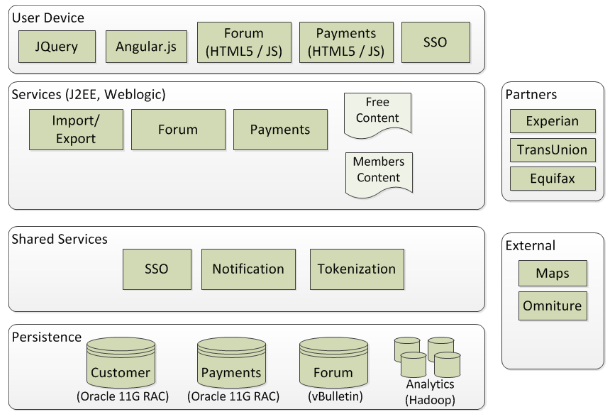
cigital

Gain Understanding From Interviews

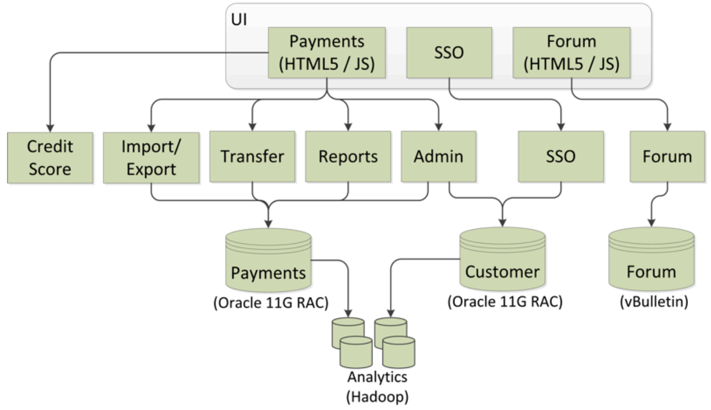
- Social-networking payment application
- Some content is free and there is membership-only content
- Some features are free and others are membership-only
- The app itself is a J2EE app and uses WebLogic as the J2EE container
- Web UI is built using JQuery JavaScript library
- The backend database is Oracle 11g
  - Database stores user’s preferences
  - Produces some membership-only reports
- This Web UI calls third-party REST services for user-specific content
- User connectivity uses HTTPS and so does interface to backend services

cigital

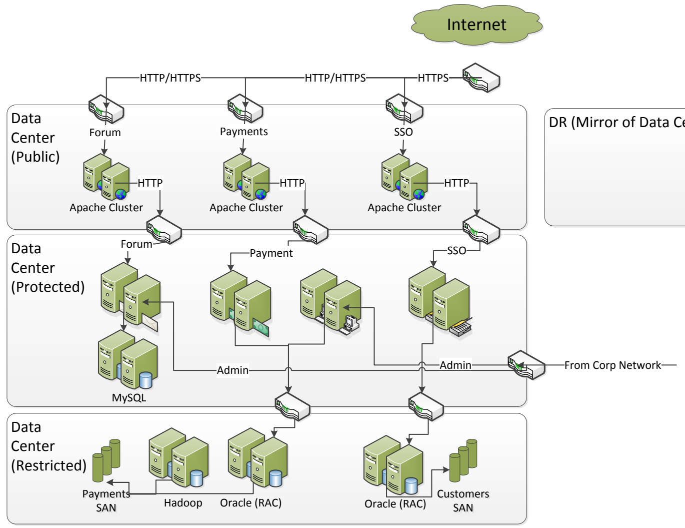
Model Diagrams



Layer Model

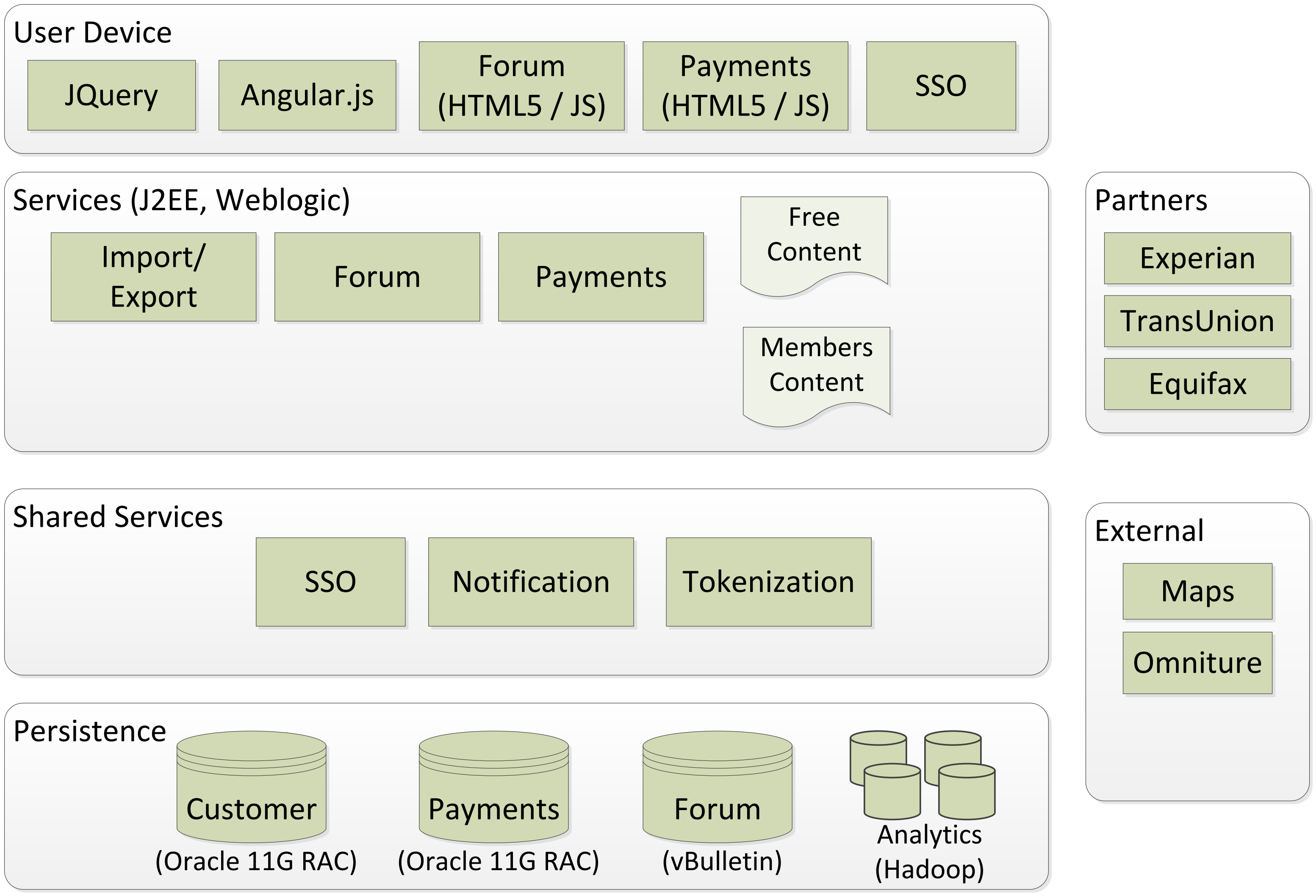


Logical Model



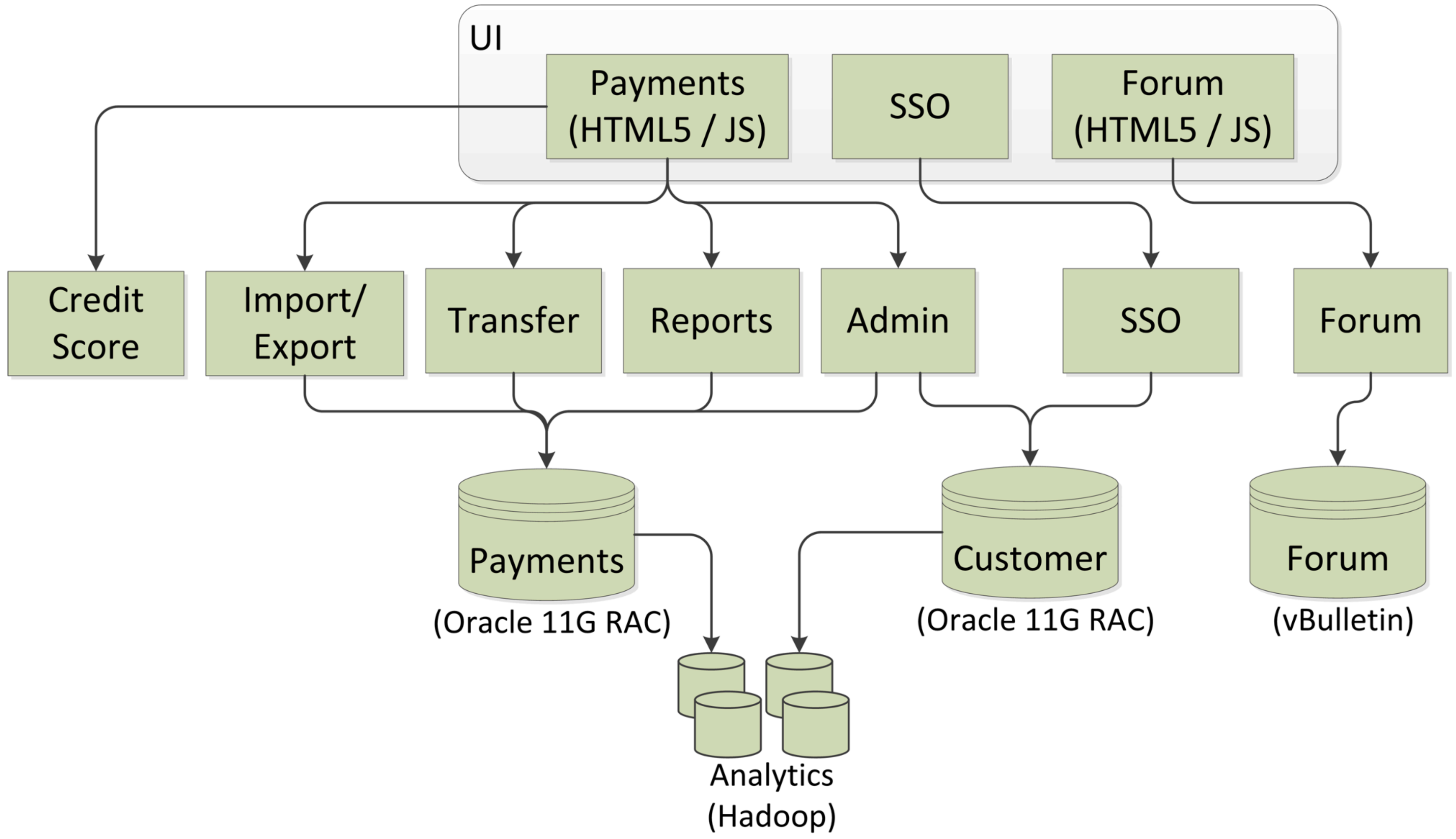
Deployment Model

Layer Model



cigital

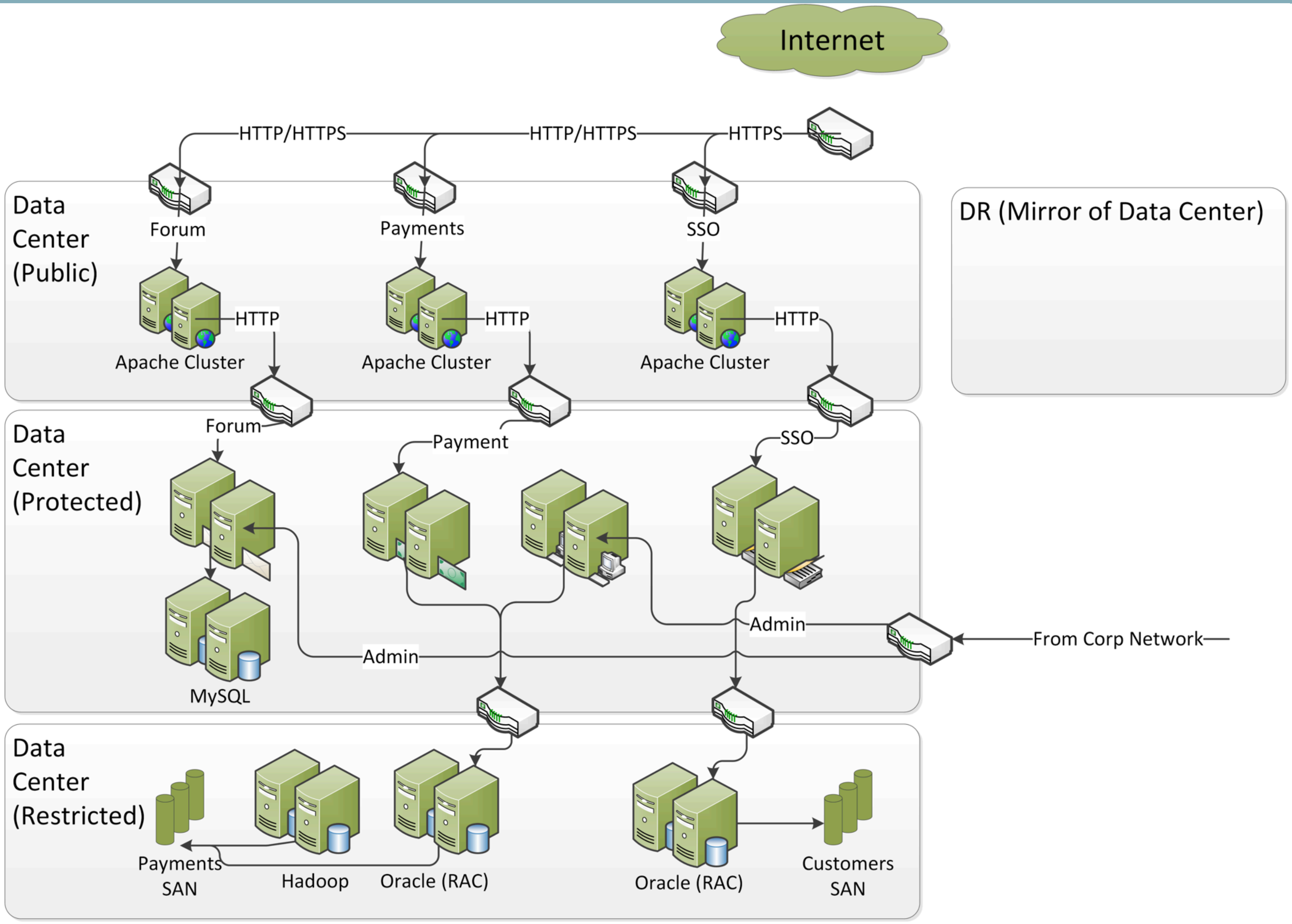
Logical Model



cigital

|

Deployment Model



Modeling The System Structure

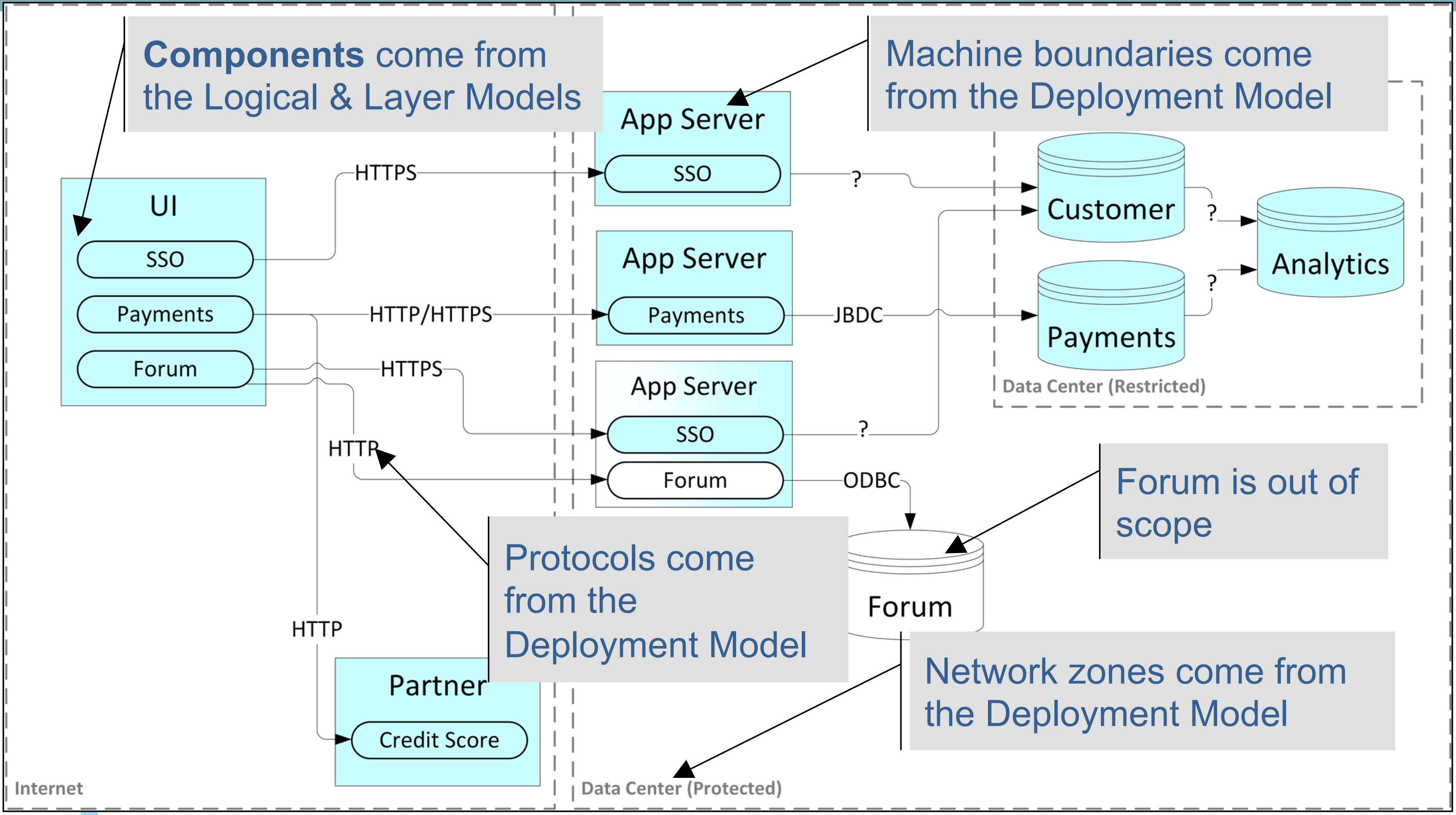
Based on interviews and diagrams, create a model that captures:

- The components of the system that are in-scope for this "release"
- How control flows between the in-scope components
- How those components and flows relate to the host boundaries and network zones
- The application layer communication protocols connecting the components

This model can use an existing model diagram or one you create

- For this in-class example, we'll create our own to help understand the parts most relevant for a Threat Model

Simplified System Model



cigital

Modeling The Threat Structure

We continue to analyze the information we've collected in our interviews and now add the threat related elements.

Assets	The data and functions that the system must protect
Security Controls	The mechanisms currently designed and implemented to protect the Assets
Threat Agents	The actors that want to harm the system

Juxtaposing the Threat Structure and the System Model creates the actual Threat Model. Interpreting the model produces a list of potential threats.

cigital

Identifying **Assets** from Interviews

Information collected in development interviews:

- Social-networking payment application
- Some content and features are membership-only; some are free
- The app is a J2EE app; uses WebLogic as the J2EE container
- The backend database is Oracle 11g
  - Stores user's preferences
  - Produces some membership-only reports
- Web UI is built using JQuery JavaScript library
- Web UI calls third-party REST services for user-specific content
- User connectivity and interface to backend services uses HTTPS

cigital

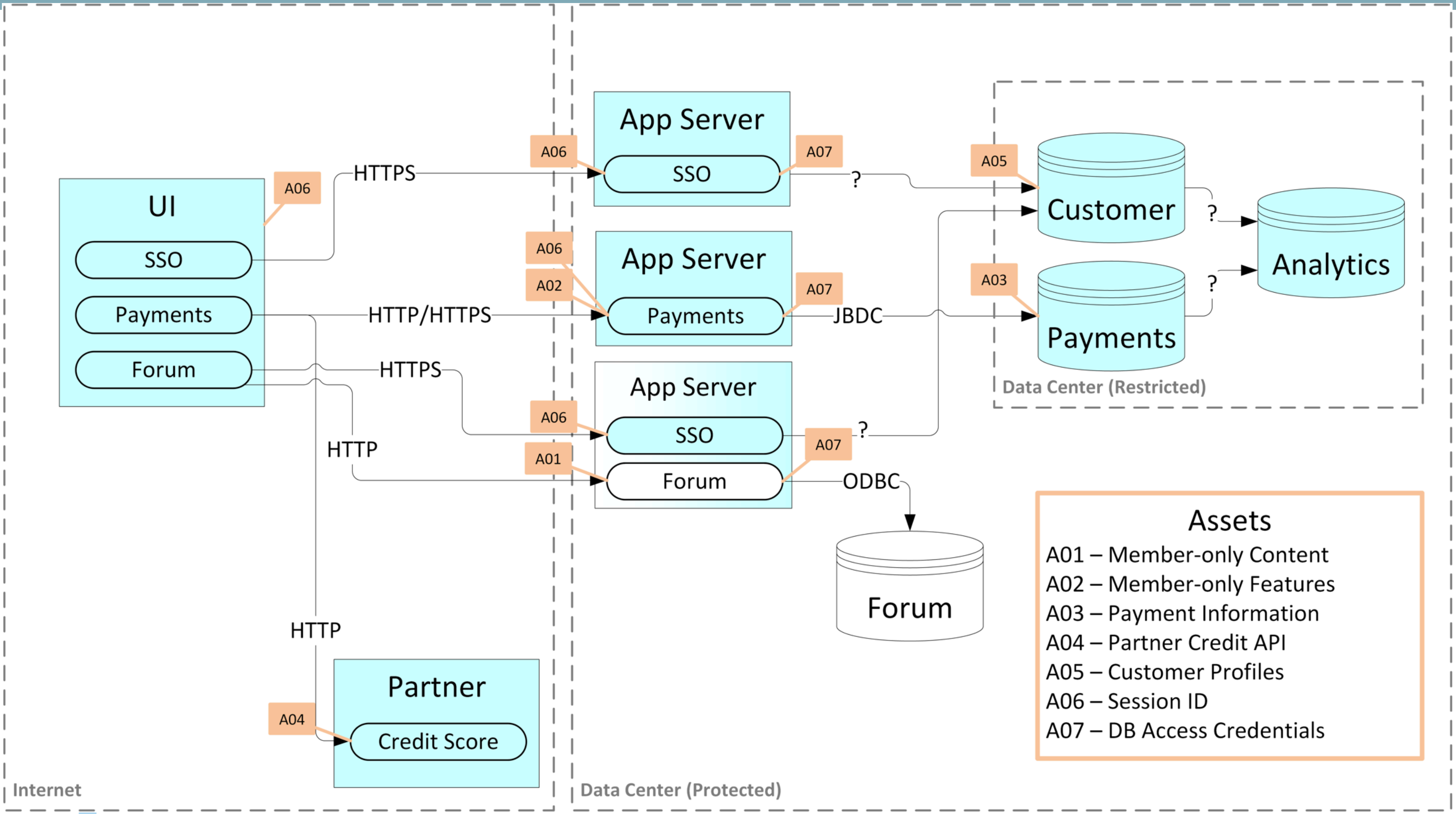
Identifying **Assets** from Interviews

Information collected in development interviews:

- Social-networking payment application
- Some content **[A01]** and features **[A02]** are membership-only; some are free
- The app is a J2EE app; uses WebLogic as the J2EE container
- The backend database **[A03]** is Oracle 11g
  - Stores user's preferences
  - Produces some membership-only reports
- Web UI is built using JQuery JavaScript library
- Web UI calls third-party REST services **[A04]** for user-specific content
- User connectivity and interface to backend services uses HTTPS

cigital

Update Model With Assets



Assets
A01 – Member-only Content
A02 – Member-only Features
A03 – Payment Information
A04 – Partner Credit API
A05 – Customer Profiles
A06 – Session ID
A07 – DB Access Credentials

Identifying Controls from Interviews

Information collected in development interviews:

- Social-networking payment application
- Some content and features are membership-only; some are free
- The app is a J2EE app; uses WebLogic as the J2EE container
- The backend database is Oracle 11g
    - Stores user's preferences
    - Produces some membership-only reports
- Web UI is built using JQuery JavaScript library
- Web UI calls third-party REST services for user-specific content
- User connectivity and interface to backend services uses HTTPS

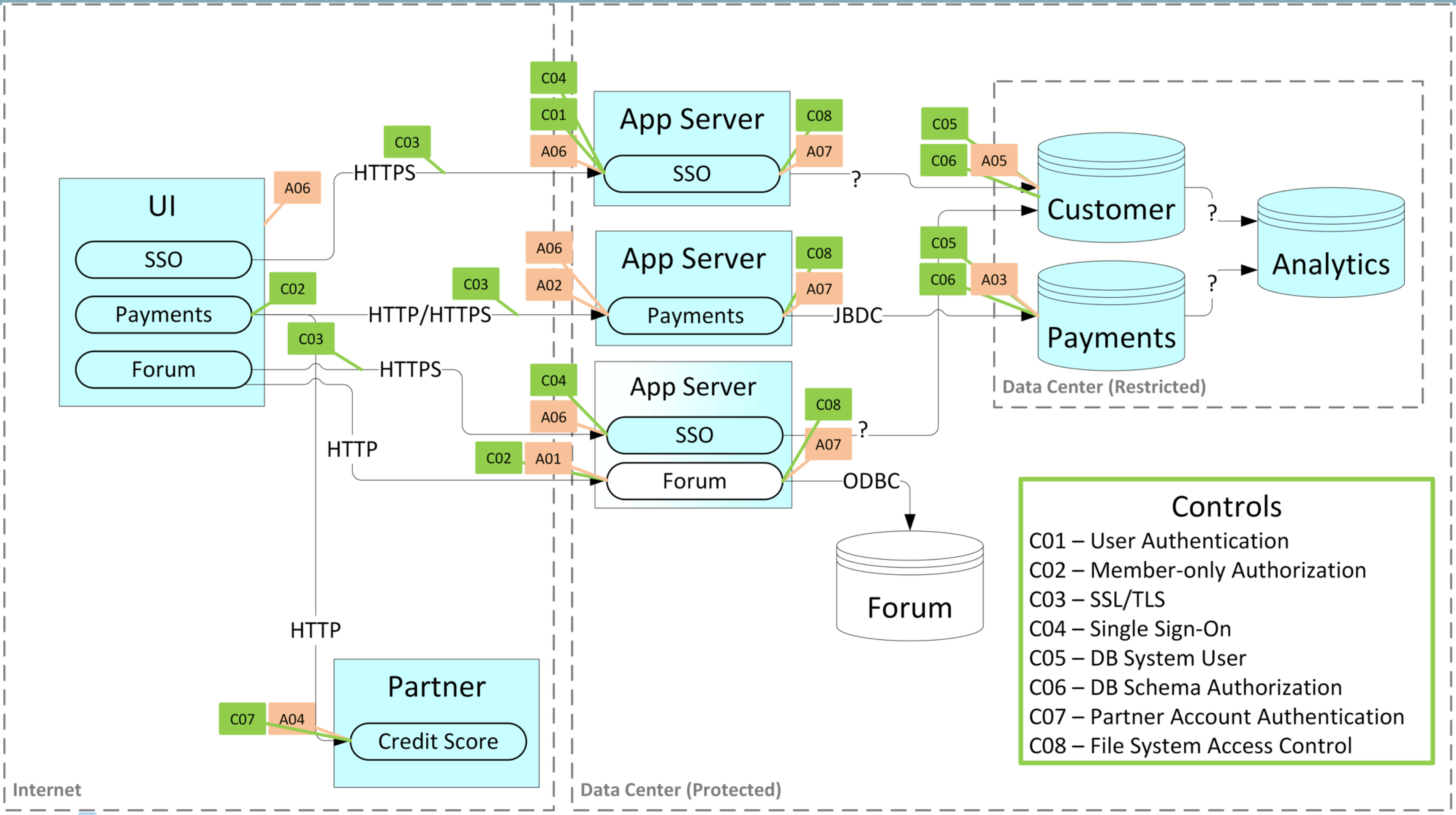
cigital

Identifying **Controls** from Interviews

Information collected in development interviews:

- Social-networking payment application
- Some content and features are membership-only **[C01][C02]**; some are free
- The app is a J2EE app; uses WebLogic as the J2EE container
- Web UI is built using JQuery JavaScript library
- Web UI calls third-party REST services for user-specific content
- The backend database is Oracle 11g
  - Stores user's preferences
  - Produces some membership-only reports
- User connectivity and interface to backend services uses HTTPS **[C03]**

Update Model With Security Controls



Controls

- C01 – User Authentication
- C02 – Member-only Authorization
- C03 – SSL/TLS
- C04 – Single Sign-On
- C05 – DB System User
- C06 – DB Schema Authorization
- C07 – Partner Account Authentication
- C08 – File System Access Control

cigital

|

Identify Threat Agents

Threat agents are primarily based on access. To identify threat agents:

- Start with the canonical threat agents for the software
- Associate the threat agent with system components they directly interact with
- Minimize the number of threat agents by treating them as equivalence classes
  - For example, assume a technically sophisticated threat agent and a script-kiddie are the same
- Assume that a threat agent can be motivated to attack the system
  - Consider motivation when evaluating likelihood

Canonical Threat Agents

Most internet-based applications can start using canonical set of threat agents:

1. External, Internet-based Attacker
2. External (client-side), LAN-based Attacker
3. External, Malicious User
4. Internal, Malicious App/System Admin

Cloud-hosted applications should account for:

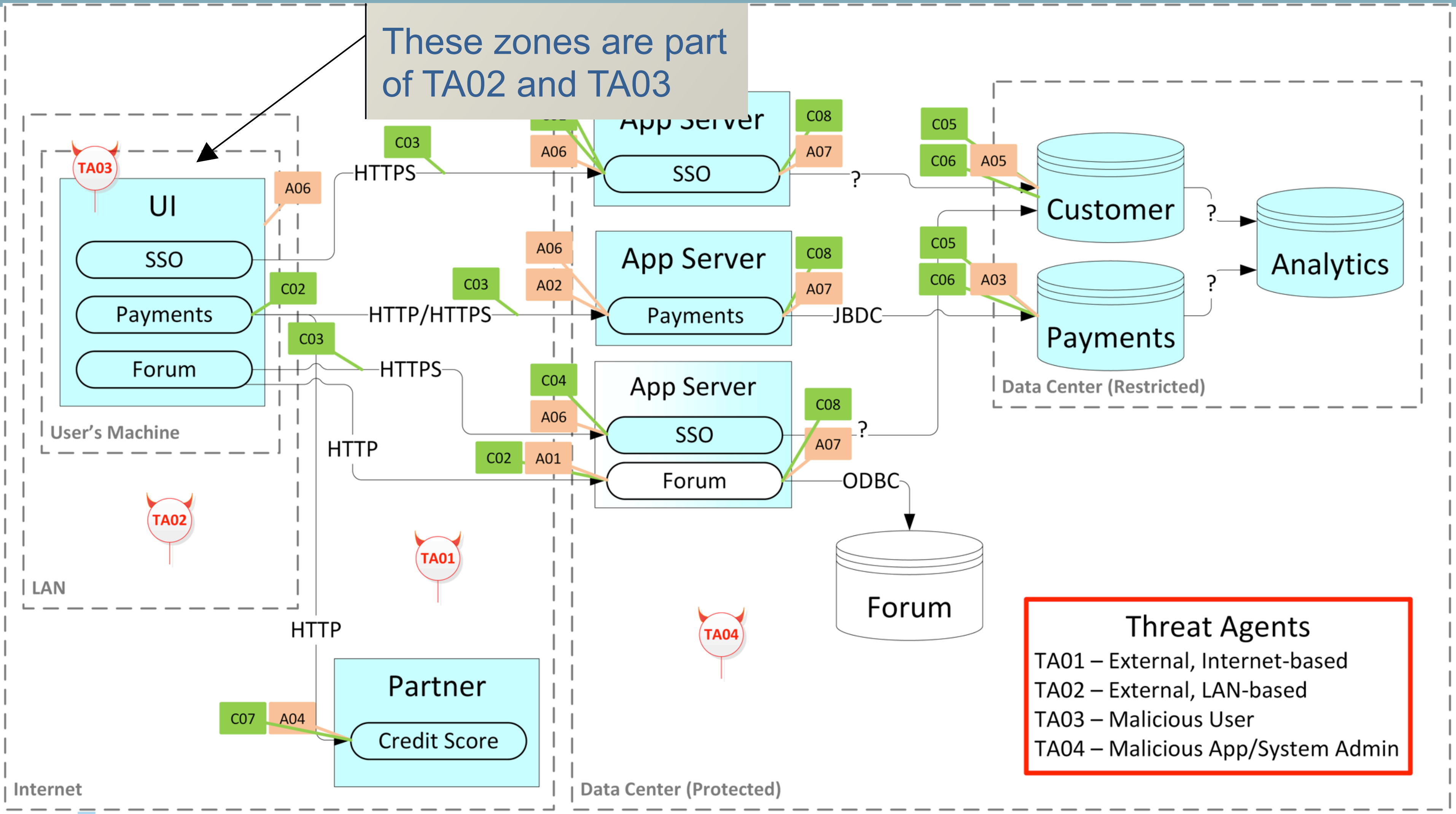
5. Malicious Cloud Provider Admin

Mobile client applications should account for:

6. Malware on a Jailbroken/Rooted device

cigital

Update Model With Threat Agents



Threat Agents
TA01 – External, Internet-based
TA02 – External, LAN-based
TA03 – Malicious User
TA04 – Malicious App/System Admin

cigital

Additional Threat Agents

Additional threat agents:

- Are business or application specific
- Generate additional potential attacks in the traceability matrix; otherwise, the threat agent is superfluous
- Increase the depth of the threat model, but also adds time to the analysis

Interpret the Threat Model

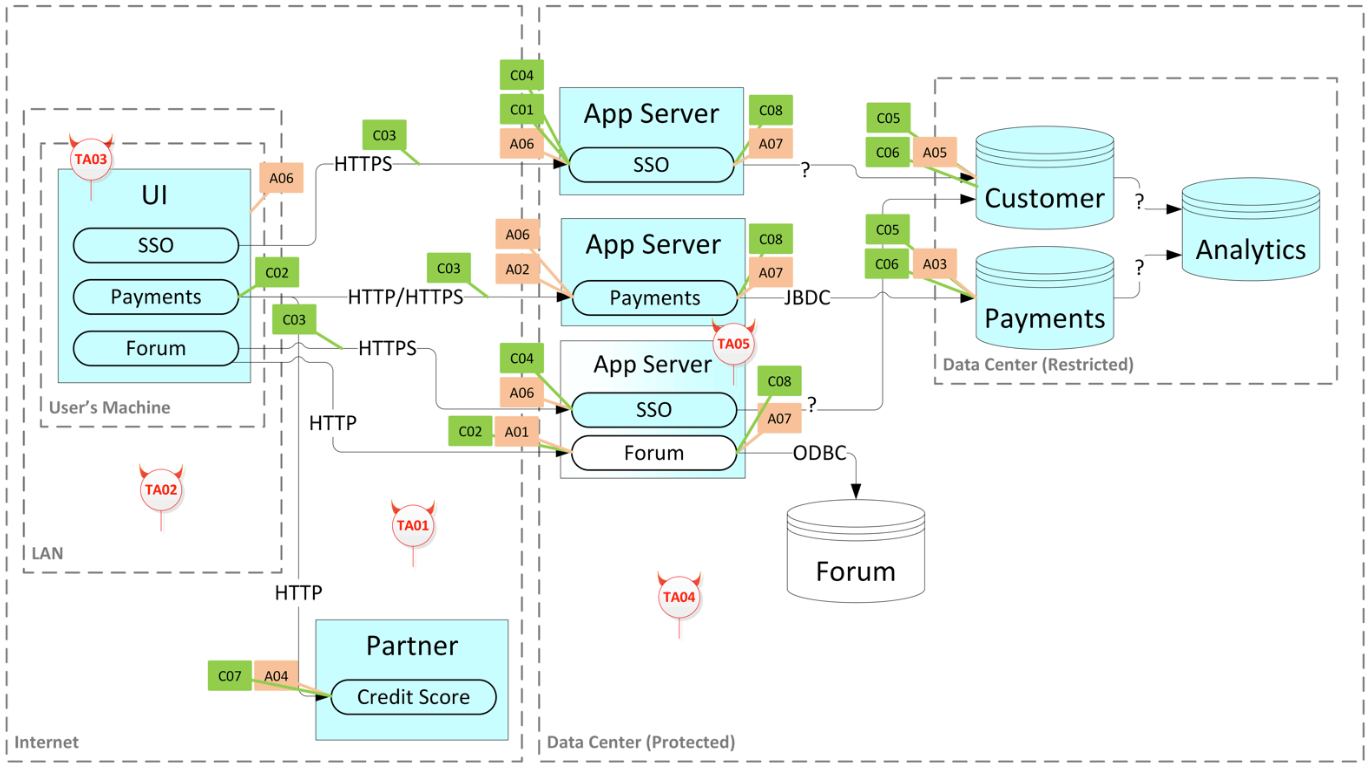
To interpret the threat model, start with threat agent and follow flow-of-control paths to reach an asset:

- Is there any path where threat agent can reach asset without going through a control?

- For any security control along each of those paths:
  - What must threat agent do to defeat the control?
  - Can threat agent defeat the control?

Record missing or weak controls in the traceability matrix

Interpret the Threat Model (In-Class)



Assets
- A01 – Member-only Content
- A02 – Member-only Features
- A03 – Payment Information
- A04 – Partner Credit API
- A05 – Customer Profiles
- A06 – Session ID
- A07 – DB Access Credentials

Threat Agents
- TA01 – External, Internet-based
- TA02 – External, LAN-based
- TA03 – Malicious User
- TA04 – Malicious App/System Admin

Controls
- C01 – User Authentication
- C02 – Member-only Authorization
- C03 – SSL/TLS
- C04 – Single Sign-On
- C05 – DB System User
- C06 – DB Schema Authorization
- C07 – Partner Account Authentication
- C08 – File System Access Control

cigital

System Threat Model Lab

System Threat Model Lab: Objectives

Lab objectives:

- Reinforce what you just learned
- Build a threat model with optional diagram for a fictitious system
- Work in independent groups.
    - Even with a defined process, people come up with different threat models
    - The models converge over time but is not likely to happen right away

cigital

System Threat Model Lab: Model the System

To model the system:

- Receive and review all artifacts
- Create a component diagram
  - OK to "flag" assets, controls, etc. in handouts
  - But just draw a component diagram now!!

Duration: 45 minutes (includes 15 min. to review)

cigital

System Threat Model Lab: Review System Models

Let’s review the system models:

- How different was each group’s interpretation of the system?

- What else would you have liked to ask in the interview(s)?

cigital

System Threat Model Lab: Add Assets & Threat Agents

Base your work **ONLY** on the provided system model diagram!

Add the following to the model:

- Assets
- Threat agents

Duration: 30 minutes (includes 10 min. to review)

cigital

System Threat Model Lab: Add Security Controls

Base your work **ONLY** on the provided system model diagram!

Add security controls to the model:

- Controls added should **ONLY** be based on documents received from client

Duration: 45 minutes (includes 20 min. to review)

|

System Threat Model Lab
Part 4: Identify Threats!

Base your work on ONLY the System Model provided

Interpret the model and construct the Traceability Matrix

- Start with a Threat Agent
- Is there any path where Threat Agent can reach Asset without going through a Control?
- For any Security Control along each of those paths:
  - What must the Threat Agent do to defeat the Control?
  - Can Threat Agent defeat the Control?

Duration: 30 minutes (includes 10 min. to review)

cigital

Thank You