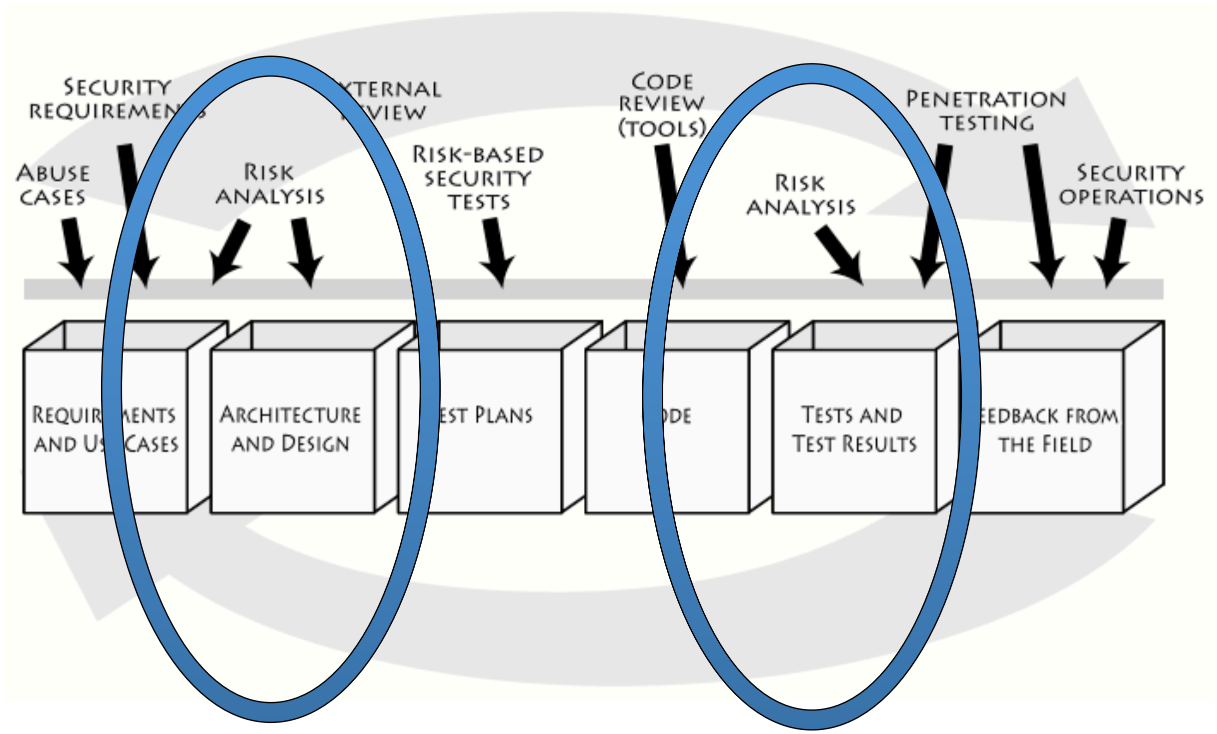
Threat Modeling

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

Threat Model Process

Define scope and depth of analysis

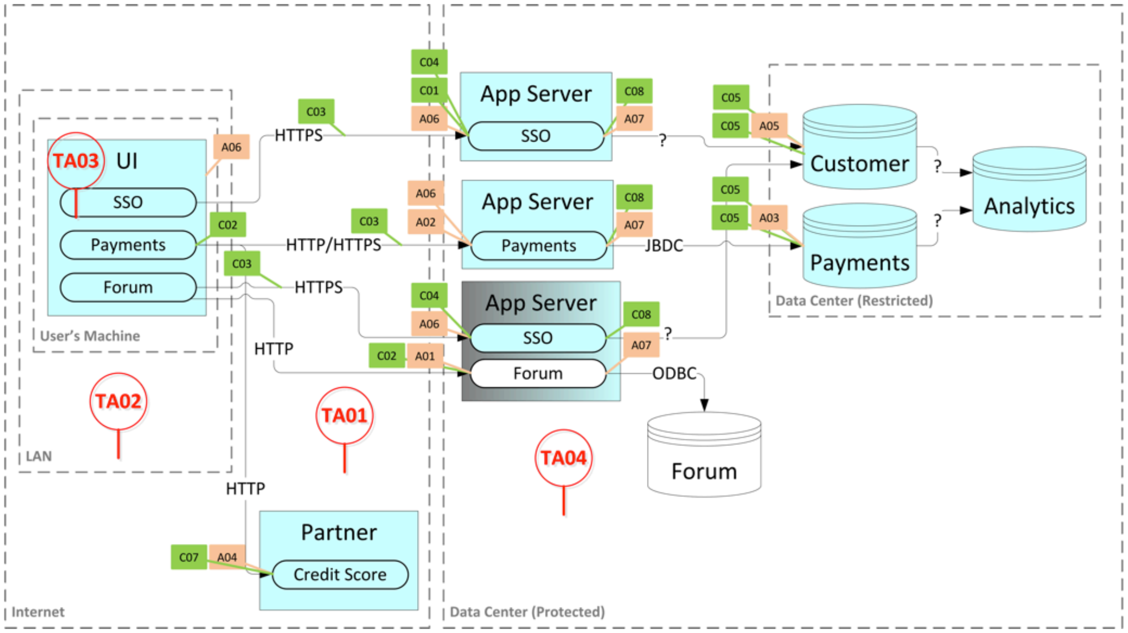
Gain understanding of what is being modeled

Model the threat structure

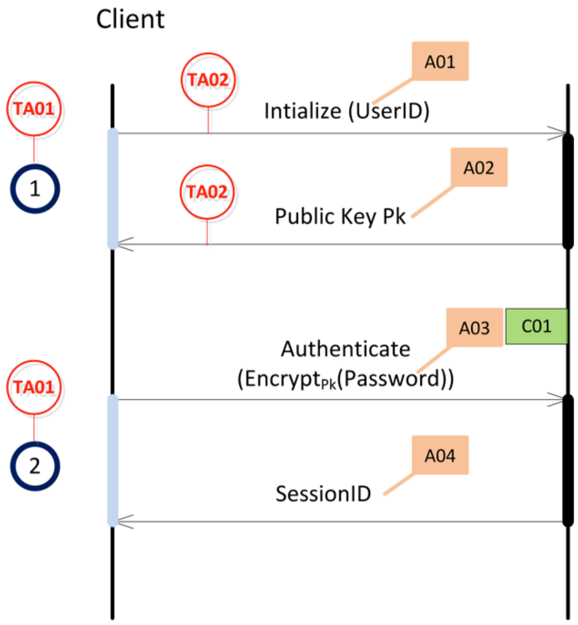
Interpret the threat model

Create the Traceability Matrix

Different Types Of Threat Models



System
Threat Model



Protocol/API
Threat Model

System Threat Models

Decompose And Model The System

Gain an understanding of how the system works

- ○ Who uses the system
- ○ Business goals/risks
- ○ Dependencies in system

Review development documentation

Interview members of the dev team

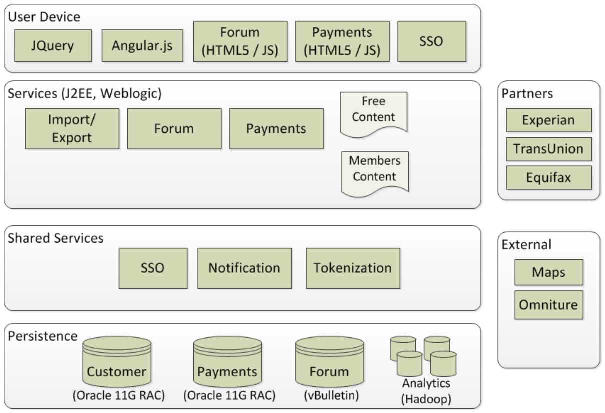
cigital

Gain Understanding From Interviews

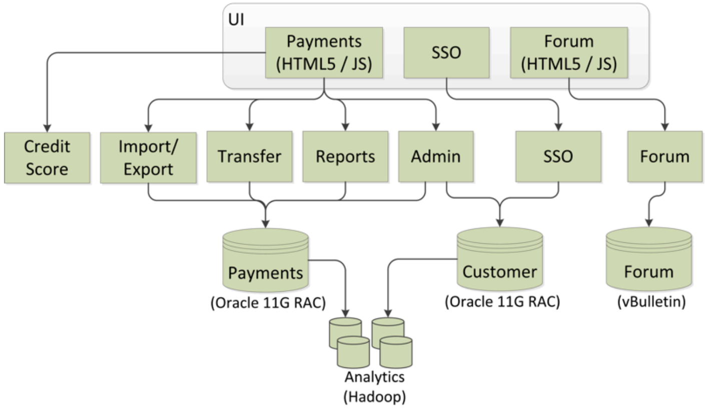
- Social-networking payment application
- Some content is free and there is membership-only content
- Some features are free and others are membership-only
- The app itself is a J2EE app and uses WebLogic as the J2EE container
- Web UI is built using JQuery JavaScript library
- The backend database is 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

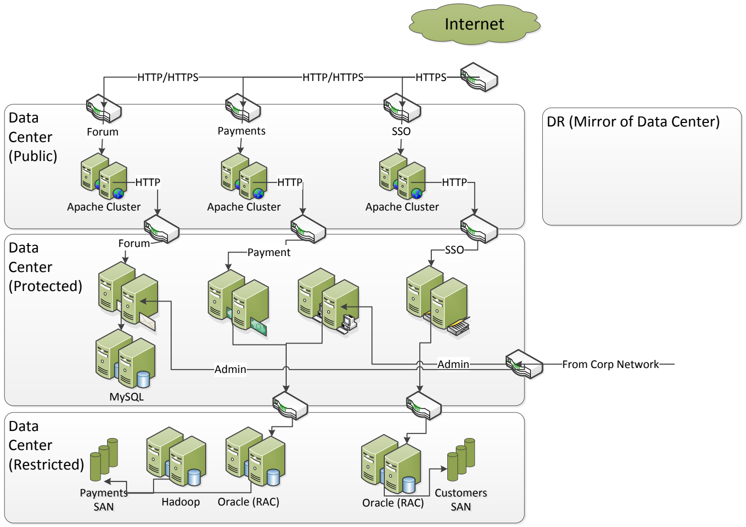
Model Diagrams



Layer Model



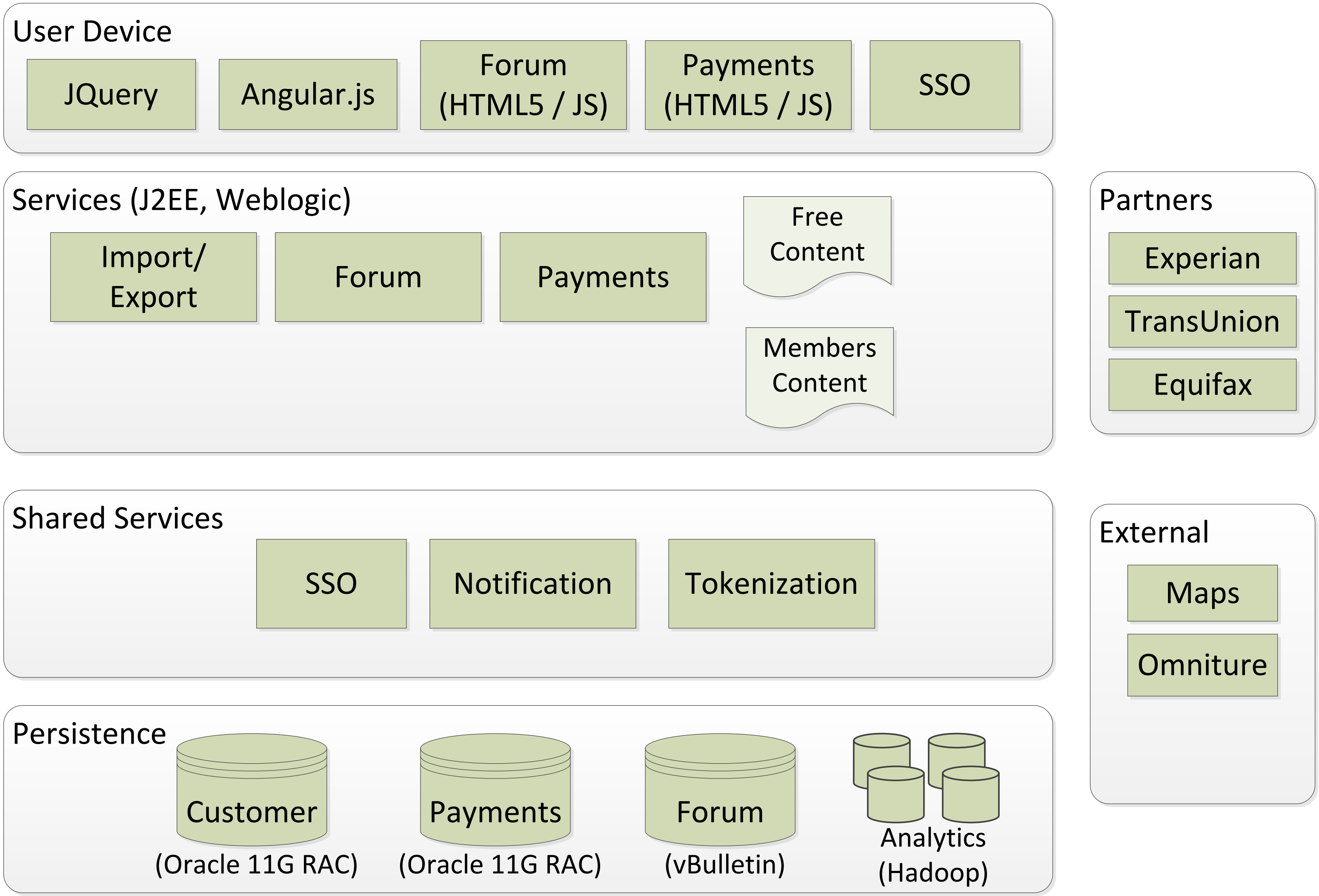
Logical Model



Deployment Model

cigital

Layer Model



User Device

JQuery

Angular.js

Forum (HTML5 / JS)

Payments (HTML5 / JS)

SSO

Services (J2EE, Weblogic)

Import/ Export

Forum

Payments

Free Content

Members Content

Partners

Experian

TransUnion

Equifax

Shared Services

SSO

Notification

Tokenization

External

Maps

Omniture

Persistence

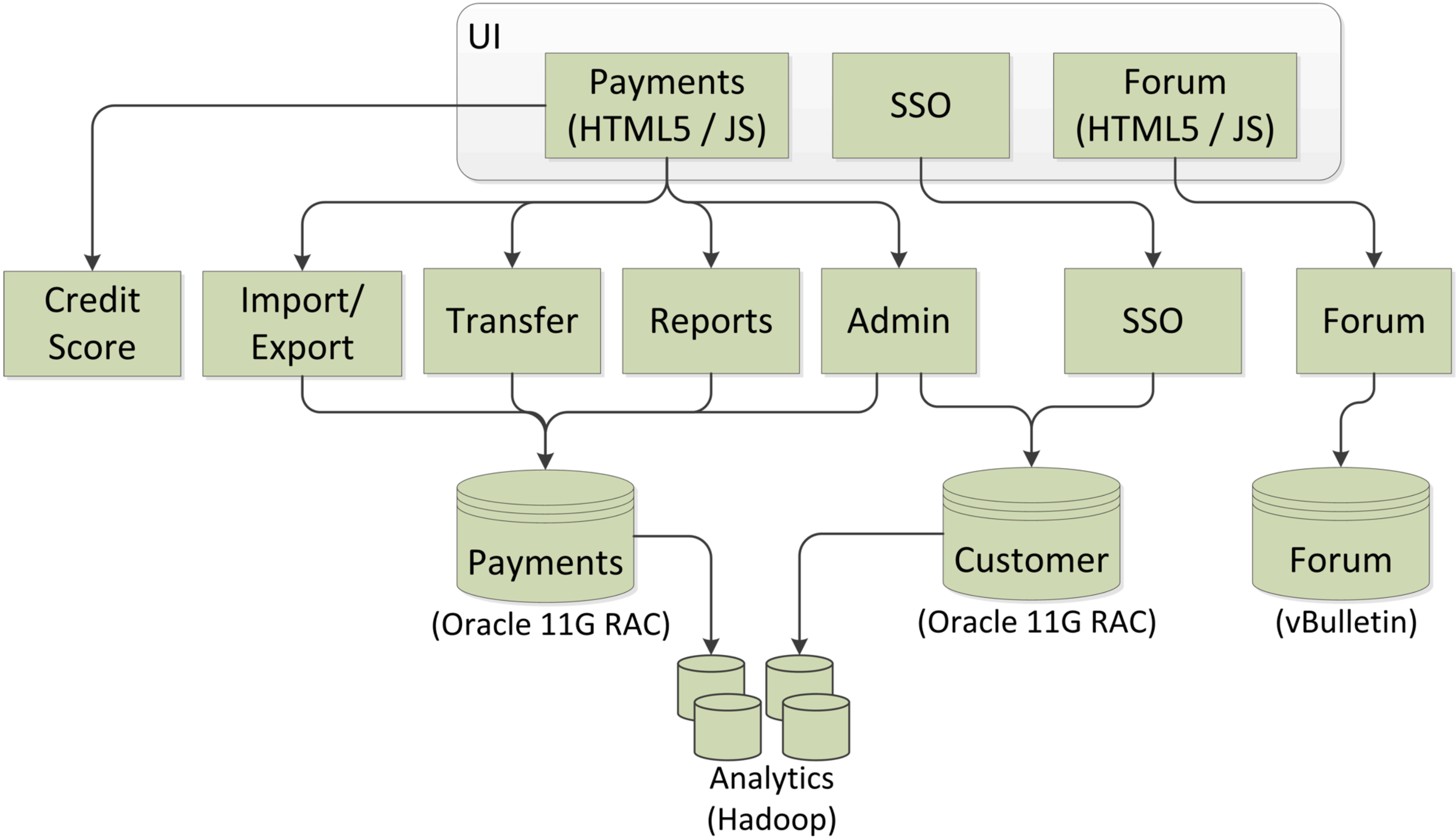
Customer (Oracle 11G RAC)

Payments (Oracle 11G RAC)

Forum (vBulletin)

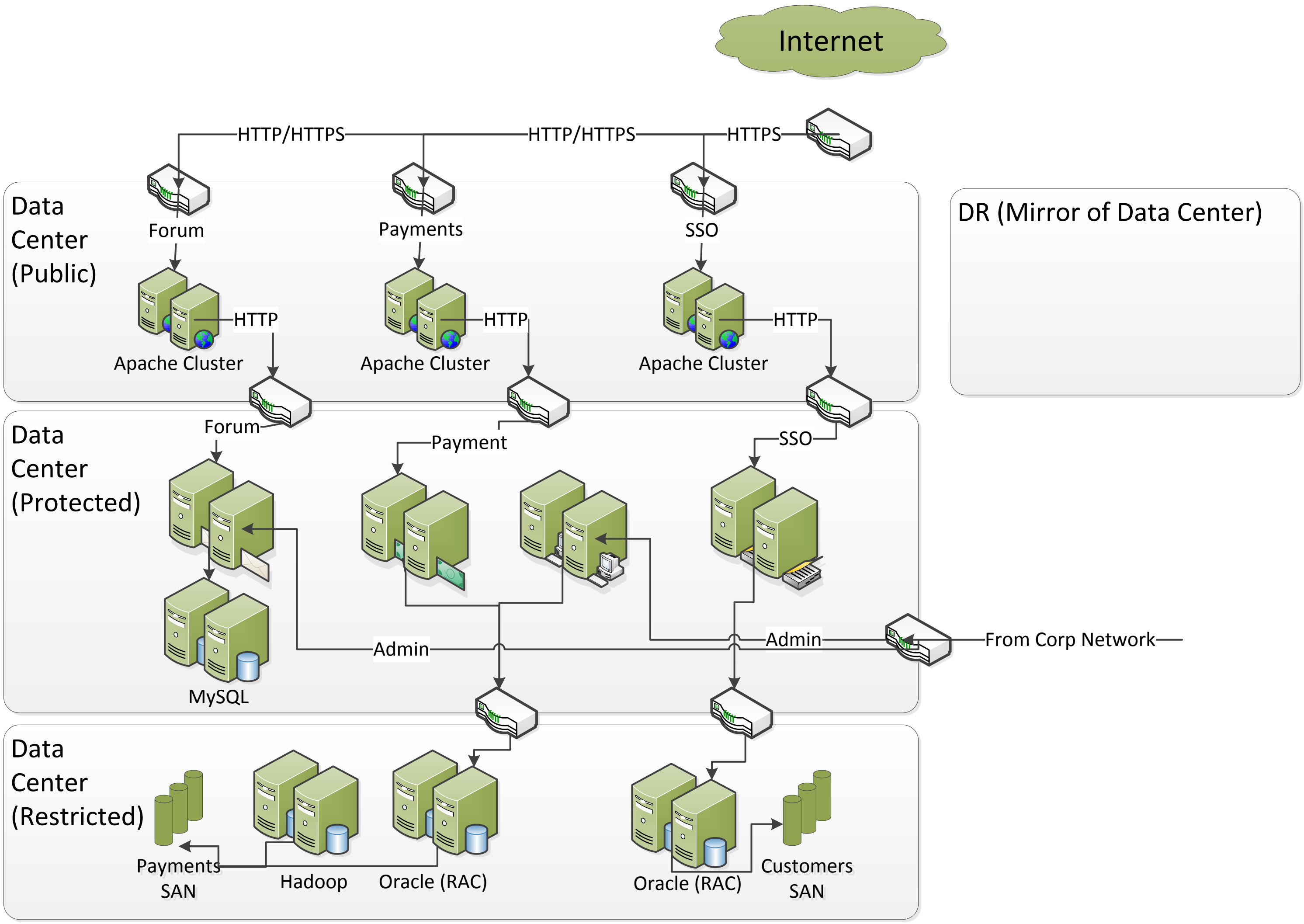
Analytics (Hadoop)

Logical Model



| | Cigital Confidential Restricted.

Deployment Model



Modeling The System Structure

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

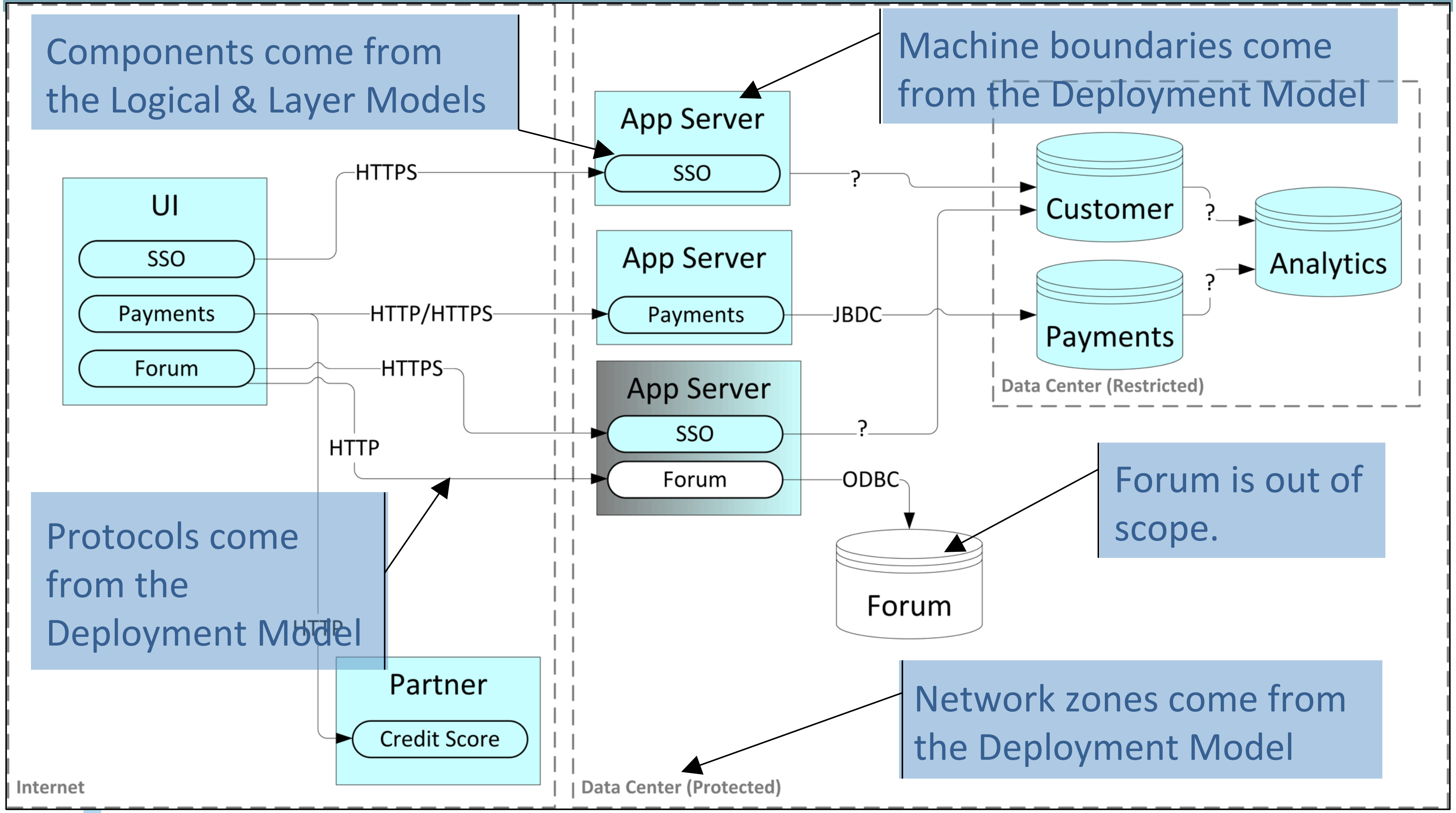
- The components of the system that are in-scope for this “release”
- How control flows between the in-scope components
- How those components and flows relate to the host boundaries and network zones
- The application layer communication protocols connecting the components

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

cigital

Simplified System Model



Components come from the Logical & Layer Models

Machine boundaries come from the Deployment Model

Protocols come from the Deployment Model

Forum is out of scope.

Network zones come from the Deployment 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.

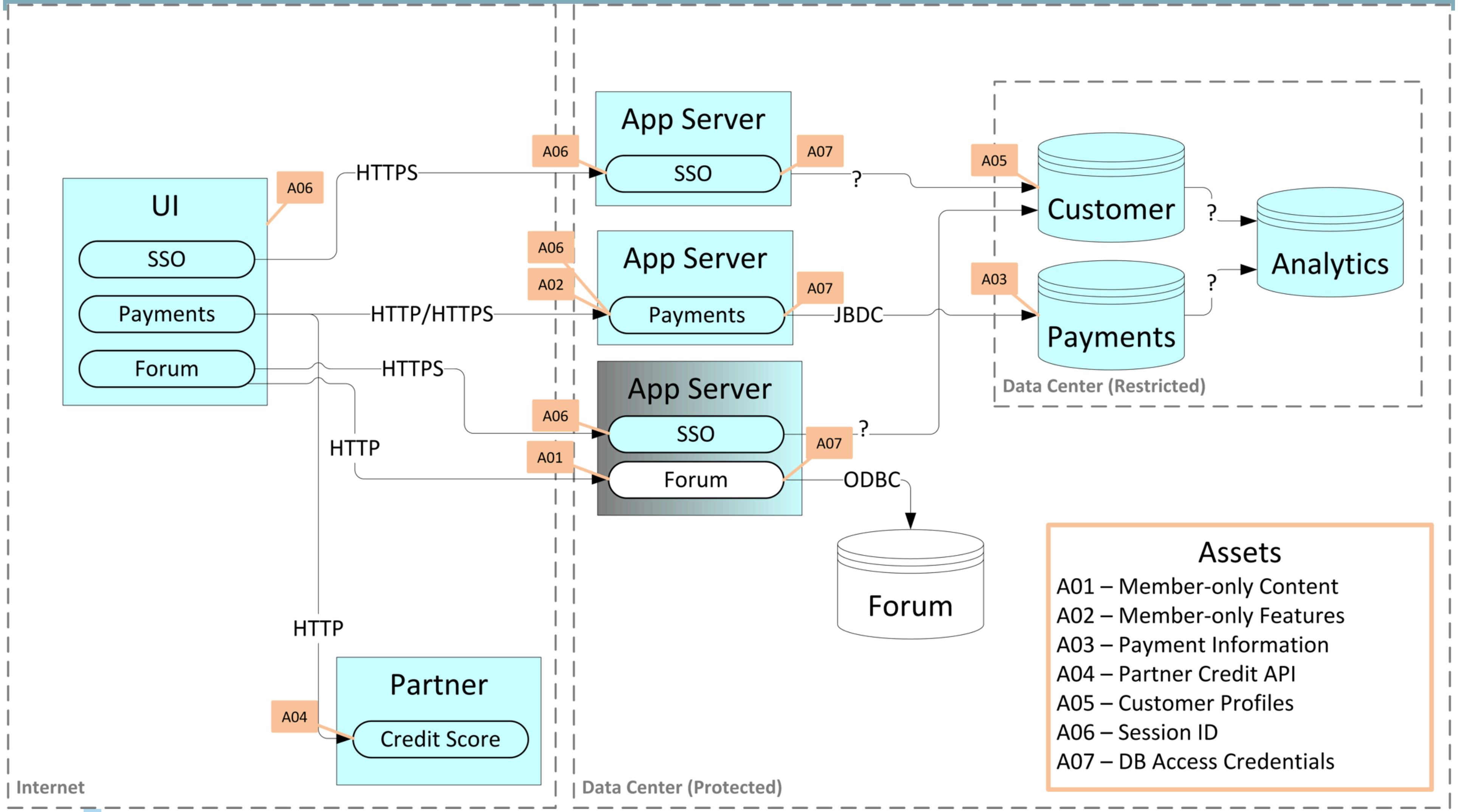
Identifying **Assets** From Interviews

- 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

Identifying **Assets** From Interviews

- Social-networking payment application
- Some content is free and there is membership-only content **[A01]**
- Some features are free and others are membership-only **[A02]**
- 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 **[A03]** is Oracle 11g
  - Database stores user's preferences
  - Produces some membership-only reports
- This Web UI calls third-party REST services **[A04]** for user-specific content
- User connectivity uses HTTPS and so does interface to backend services

Model The Threat Structure – 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

| | Cigital Confidential Restricted.

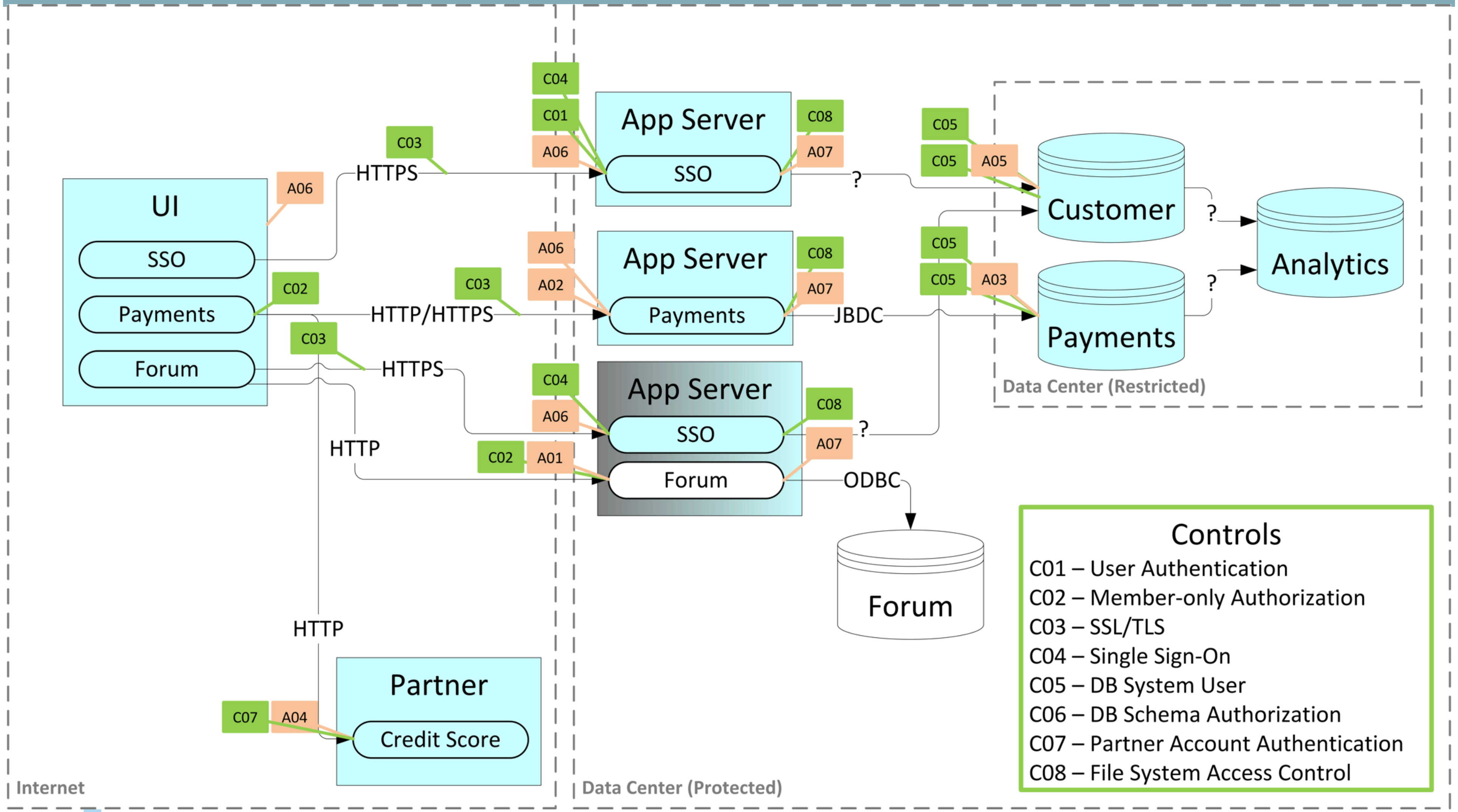
Identifying **Controls** From Interviews

- 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

Identifying **Controls** From Interviews

- Social-networking payment application
- Some content is free and there is membership-only **[C01] [C02]** content
- Some features are free and others are membership-only **[C01] [C02]**
- 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 **[C03]** and so does interface to backend services

Model The Threat Structure – 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

Identify Threat Agents

- Threat Agents are primarily based on access.
- Start with the Canonical Threat Agents for the software.
- Associate the Threat Agent with system components they can directly interact with.
- Minimize the number of Threat Agents, by treating them as equivalence classes. For example, assume a technically sophisticated attacker and a script-kiddie are the same.
- Assume that an attacker can be motivated to attack the system. Consider motivation when evaluating Likelihood.

System Threat Model Canonical Threat Agents

Most Internet-based applications can start using canonical set of Threat Agents:

- External, Internet-based Attacker
- External (client-side), LAN-based Attacker
- External, Malicious User
- Internal, Malicious App/System Admin

Cloud-hosted applications should account for:

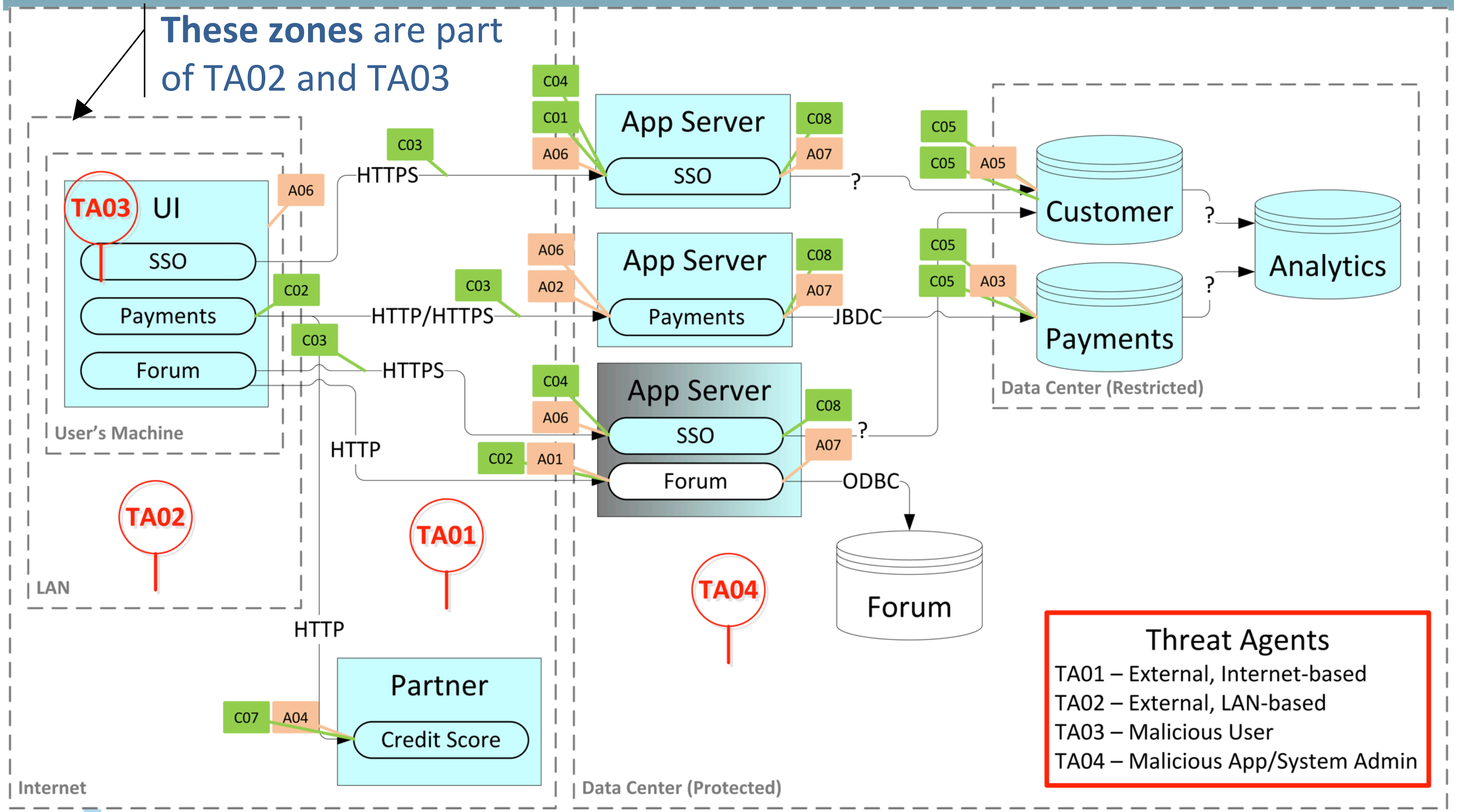
- Malicious, Cloud provider Admin

Mobile client applications should account for:

- Attacker with a jail-broken/rooted device

cigital

Model The Threat Structure – Threat Agents



These zones are part of TA02 and TA03

Threat Agents

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

cigital

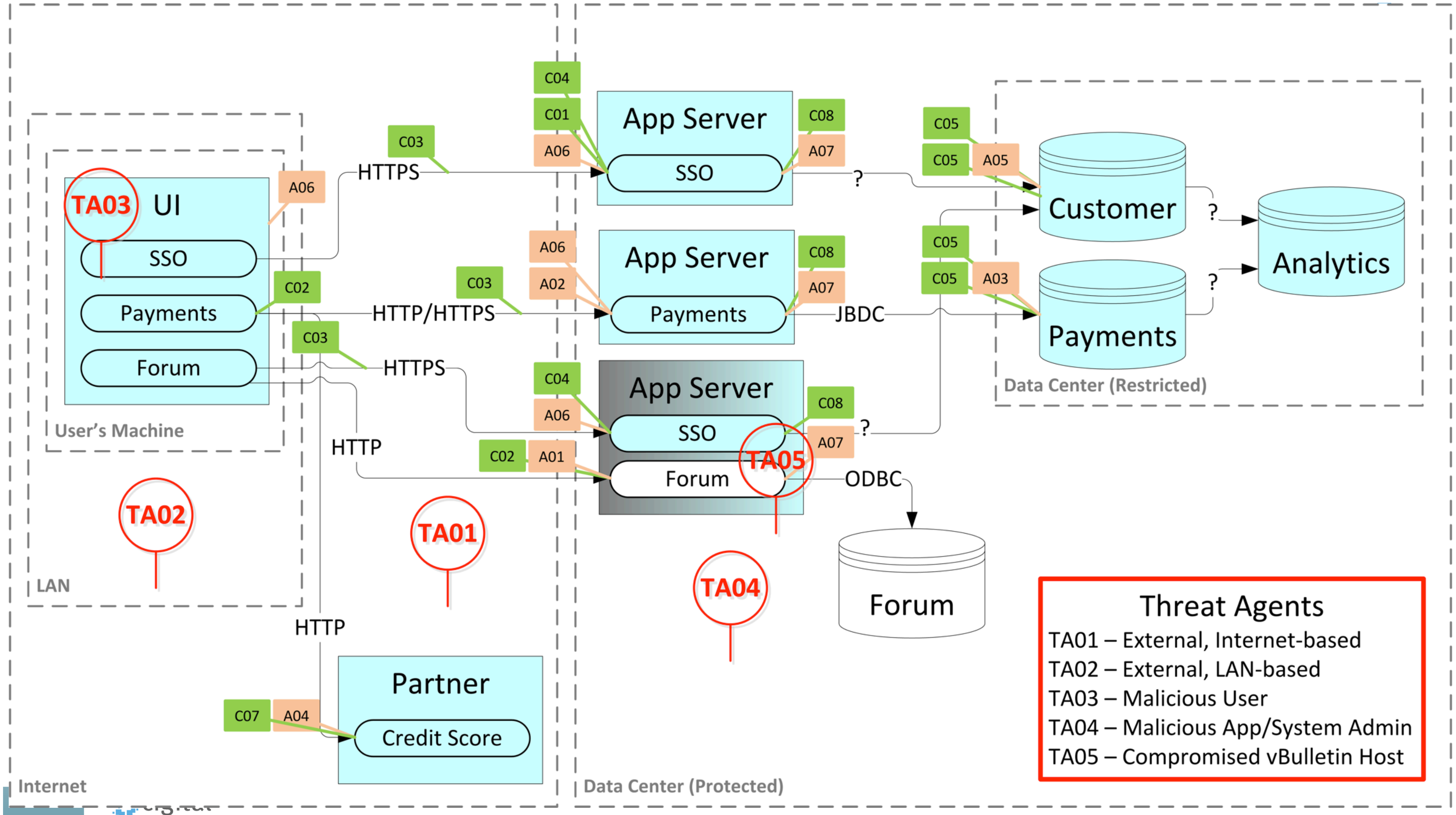
Additional Threat Agents

- Additional Threat Agents are business or application specific

- Additional Threat Agents should generate additional threats in the Traceability Matrix; otherwise, the Threat Agent is superfluous

- Additional Threat Agents increases the depth of the TM, but also adds time to the analysis

Evaluating Pivots Using Threat Agents



Threat Agents

TA01 – External, Internet-based
TA02 – External, LAN-based
TA03 – Malicious User
TA04 – Malicious App/System Admin
TA05 – Compromised vBulletin Host

Interpret The Threat Model

Using the model, start with a Threat Agent and follow the flow-of-control paths in the system to reach an Asset

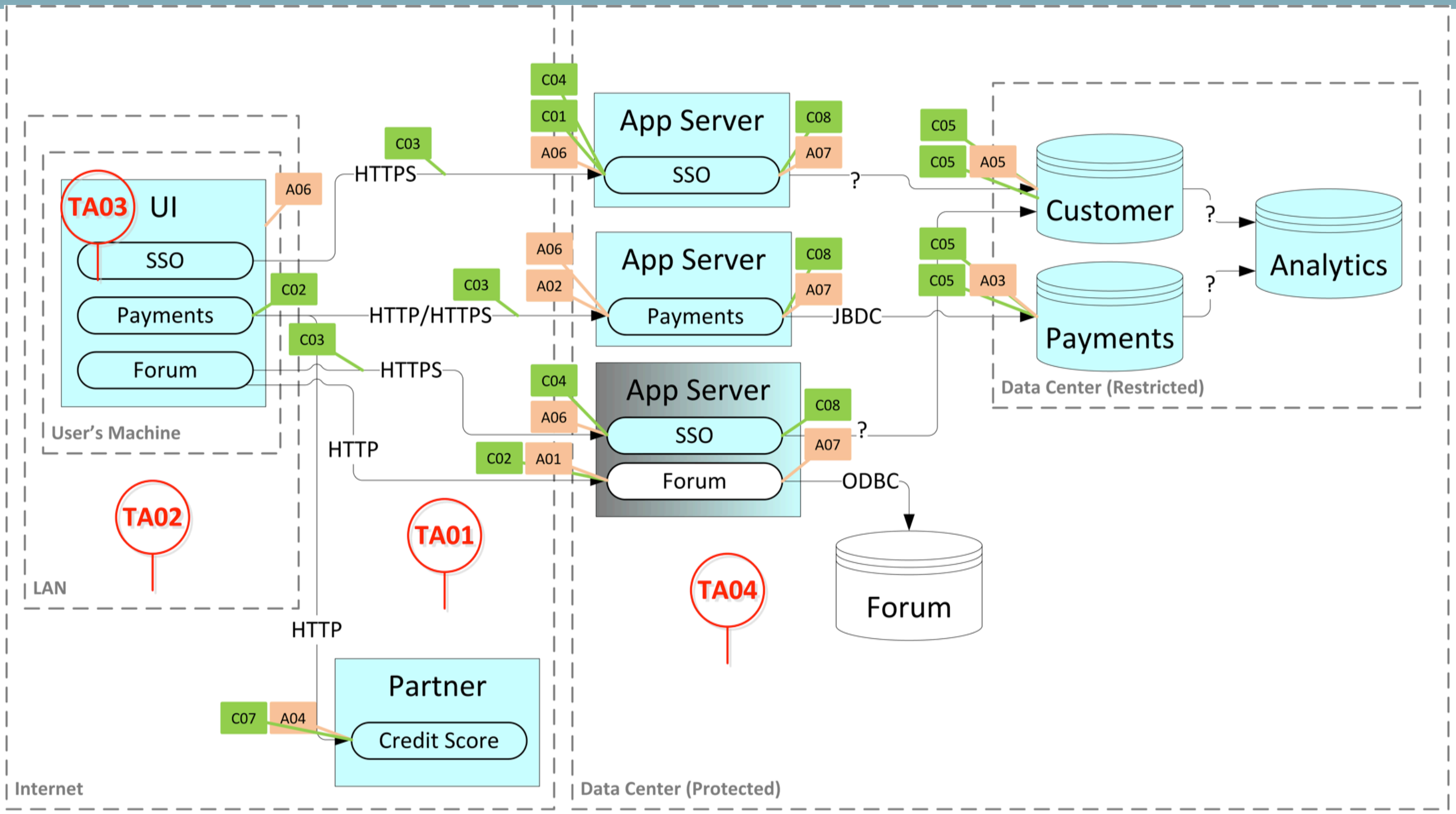
- Is there any path where Threat Agent can reach Asset without going through a Control?

- For any Security Control along each of those paths:
  - What must the Threat Agent do to defeat the Control?
  - Can Threat Agent defeat the Control?

Record missing or weak controls in the Traceability Matrix

cigital

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

Create The Traceability Matrix

Collect Threats in the Traceability Matrix.

Each entry in the Traceability Matrix:

- Identifies a threat
- Calculates the risk based on the Threat Agent and the existing controls
- Proposes mitigations to development to reduce the risk to an acceptable level
  - Mitigations should be practical and implementable
  - Important to create a “shared vision” with the development team

Traceability Matrix Entry

Threat Agent

Asset

Attack

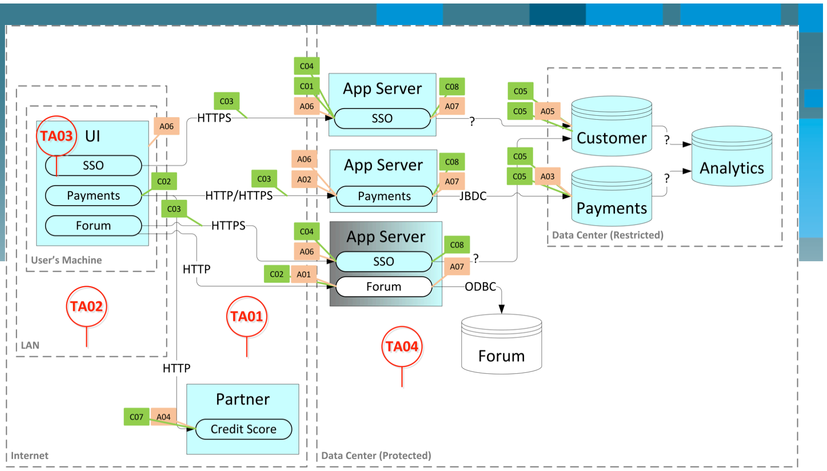
Attack Surface

Attack Goal

Impact

Security Control

Populated Traceability Matrix



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

Threat Agent	Asset	Attack	Attack Surface	…	Mitigation

System Threat Model Lab

System Threat Model Lab – Objectives

Reinforce what you just learned

Build a complete threat model with optional diagram for a fictitious System

Work in independent groups and understand that even with a defined process, different people come up with different threat models

System Threat Model Lab
Part 1: Model The System

Receive and review all artifacts

Review interview notes about the system

Create a component diagram

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

cigital

System Threat Model Lab
Part 1: Review System Models

How different was each group's interpretation of the System?

System Threat Model Lab
Part 2: Add Assets & Threat Agents

Base your work on ONLY the System Model diagram provided!!

Add Threat Structure to the Model:

- Assets
- Threat Agents

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

System Threat Model Lab
Part 3: Add Security Controls

Base your work on ONLY the System Model provided!!

Add Threat Structure to the Model:

- Security Controls

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

cigital

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)

Thank You

cigital