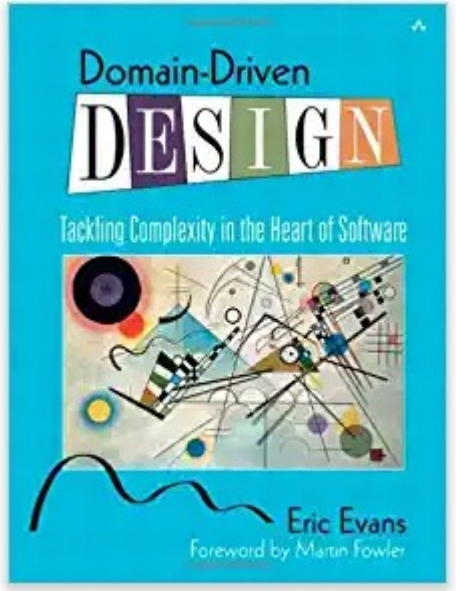
Domain Driven Design

> Domain Driven Design approach provides principles and patterns to address the challenges faced with developing complex domain models



Domain Driven Design

Objectives

- IT project's primary focus is on the core domain and domain logic
- Break a unified model into smaller more manageable models
- Iterative improvements to the model to solve a domain problem

Domain Driven Design

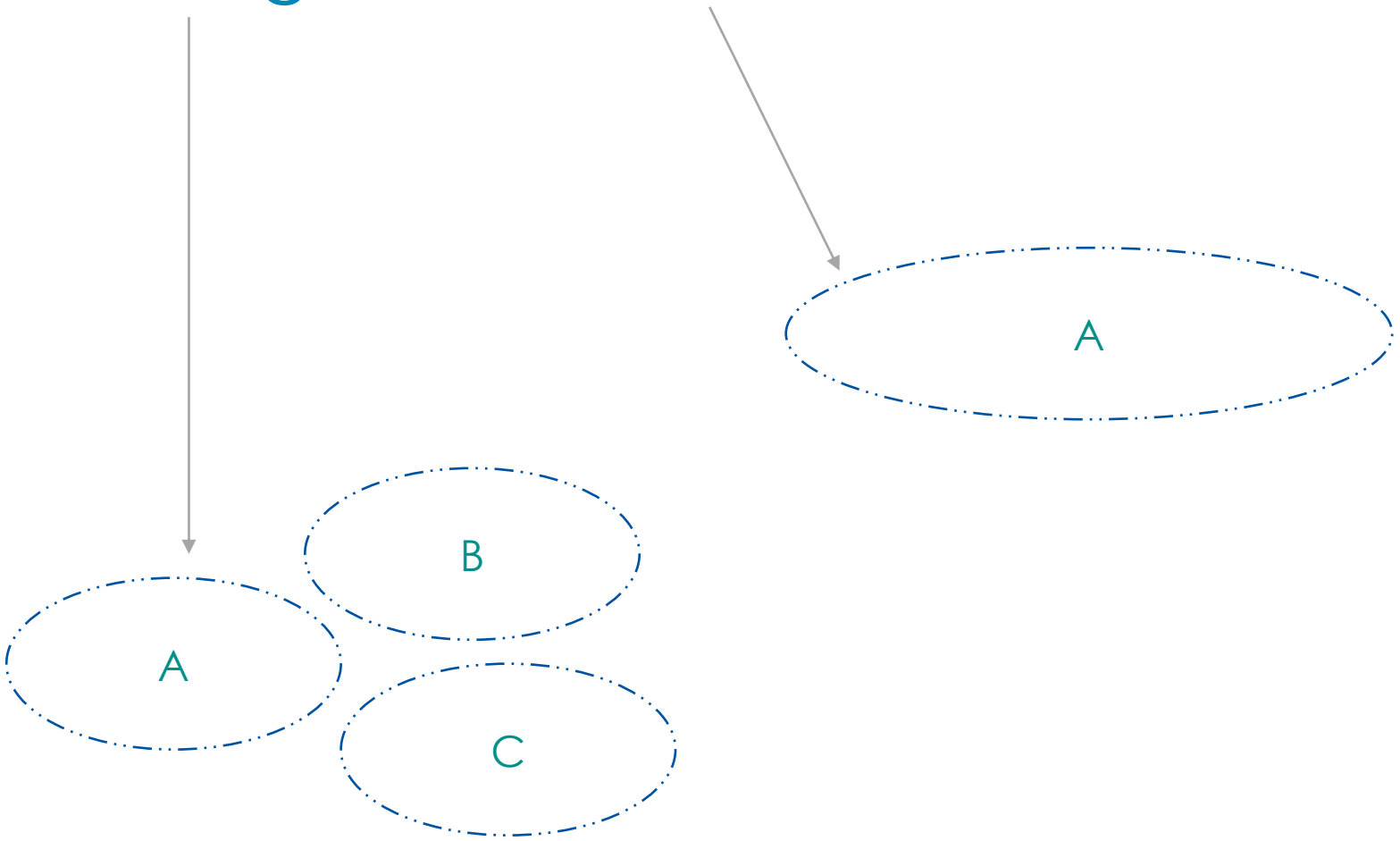
Strategic Vs. Tactical Patterns

Translate of the conceptual models to software application | services design e.g., classes, modules

Divide a complex & large business problem into smaller chunks with well defined boundaries

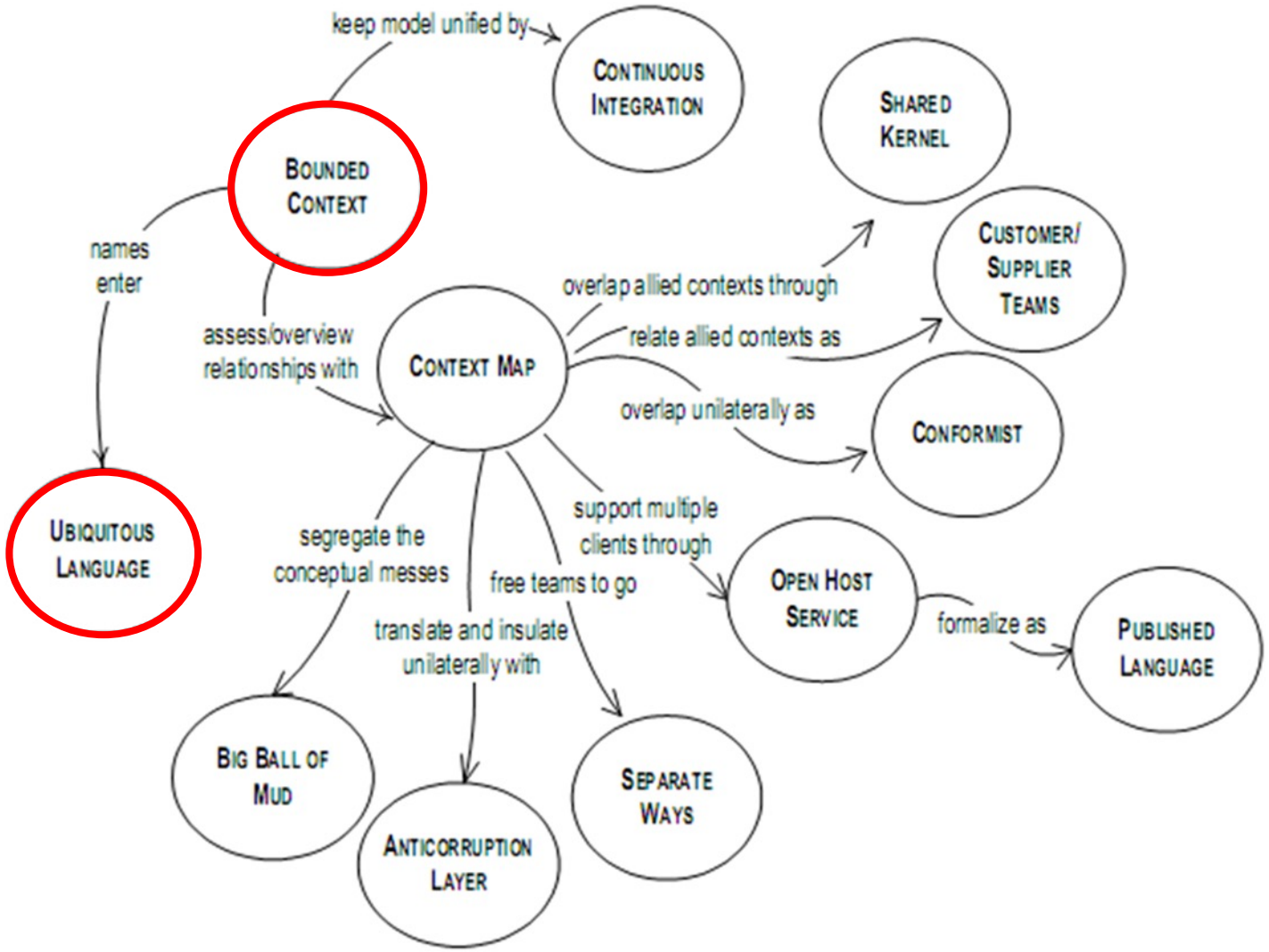
Domain Driven Design

Strategic Vs. Tactical Patterns



Domain Driven Design

Strategic Patterns



Domain Driven Design

Microservices

> Domain Driven Design approach leads to smaller independent domain models that can then be built as highly decoupled, independent set of microservices

Section objective

1. Identify and categorize the Sub-domains
2. Business context
3. DDD Strategic Pattern : Ubiquitous Language
4. DDD Strategic Pattern : Bounded Context
5. Discovering the Bounded Contexts

Introduction to DDD

Terminology

How does DDD address the challenges of Modeling?

What is a sub-domain?

…

Addressing the Challenges

> Domain Driven Design approach provides principles and patterns to address the challenges faced with developing complex domain models

Quick Review

Domain = A sphere of knowledge, influence, or activity

- Made up of MULTIPLE Sub-domains
- Multiple Domain Experts needed to support business functions

Subdomains

Understanding the business requires breaking it into parts

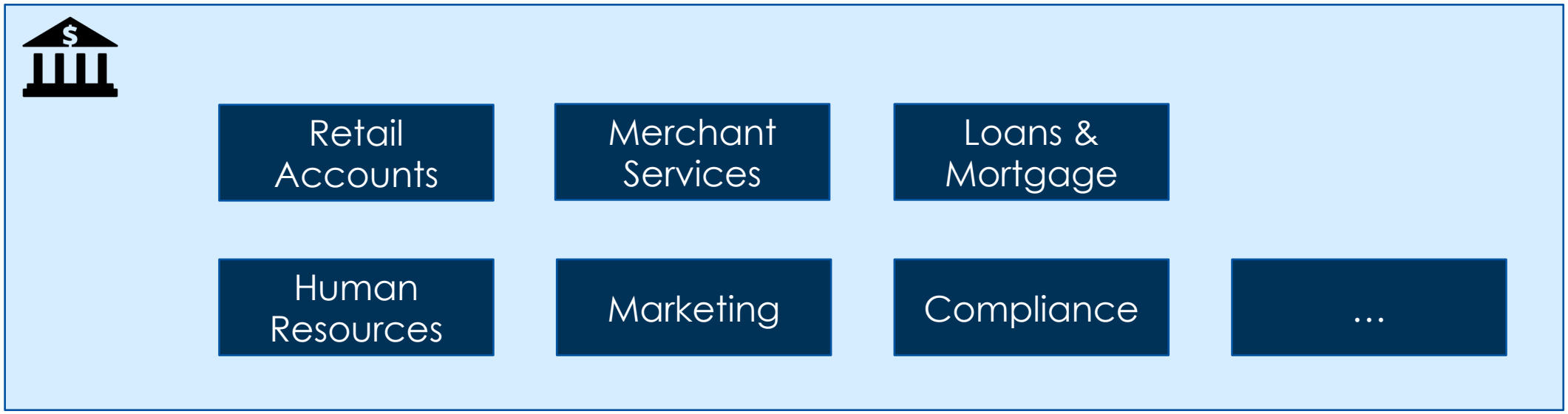
Subdomain

Types of Subdomains

Build Vs. Buy decisions

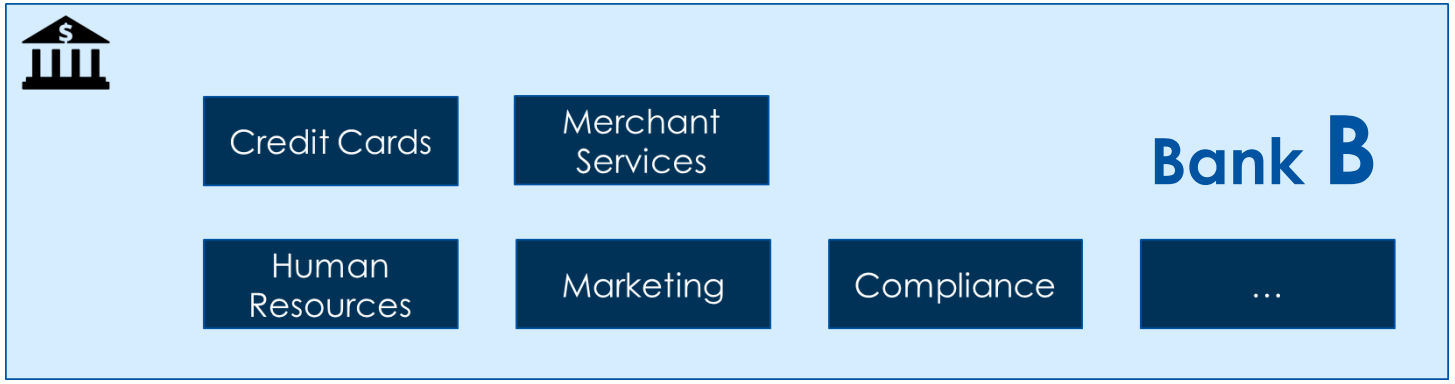
Sub-Domains

A business domain is composed of multiple subdomains



Sub-Domains are specific to the Enterprise

Business may not operate in all sub domains in that overarching domain (industry)



Sub-Domains Granularity

Depends on the focus of the business

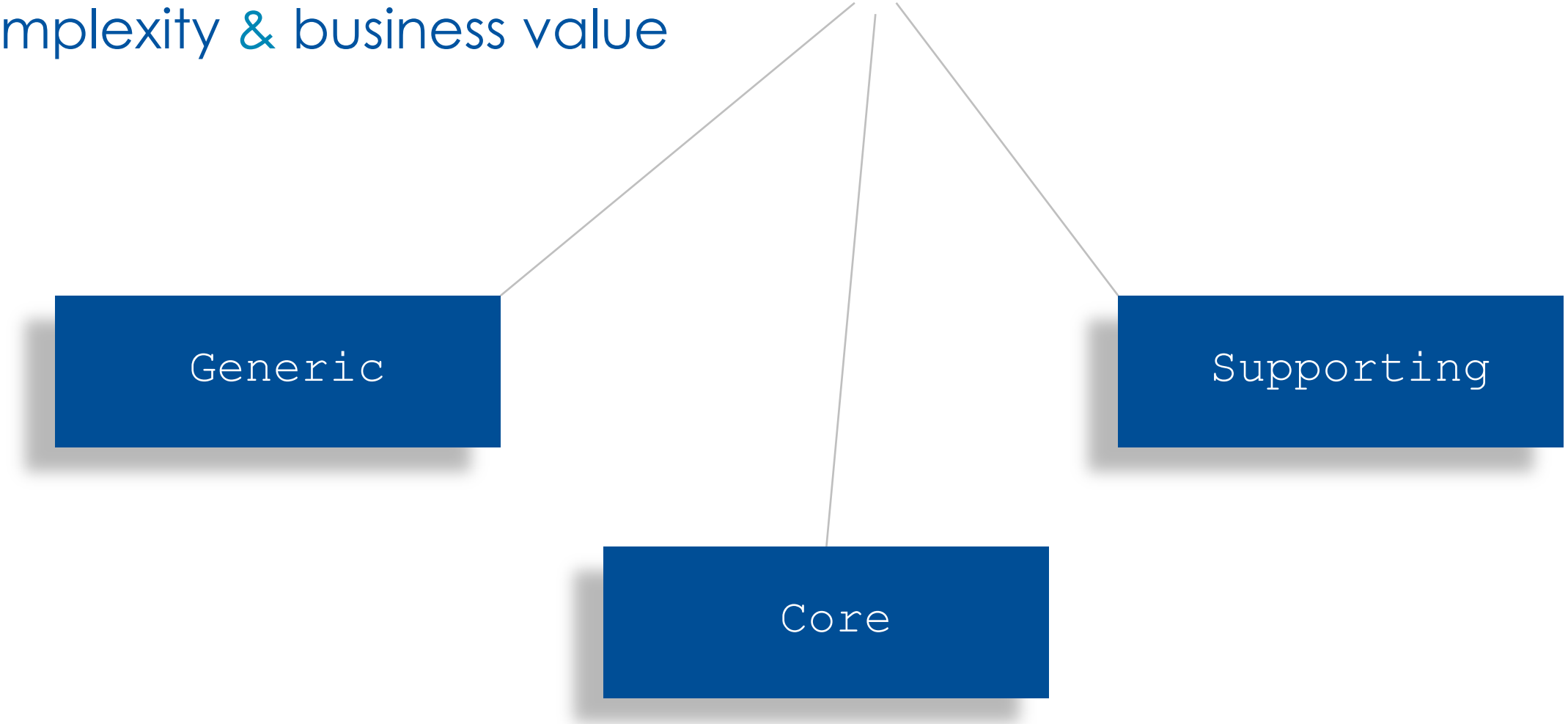
Sub-Domain Complexity

Each Sub-Domain has a different level of complexity

- Business Rules
- Compliance
- Complex Calculations
- Process & Handovers
- Dynamicity
- …

Sub-Domain Categories

Sub-domains are categorized into 3 types based on their complexity & business value



Generic Subdomain

Known solutions exist for such subdomains

Generic

- Nothing special about these subdomains
- Best practices are available
- NO business advantage in re inventing the wheel

Human Resources

Facilities Management

Core Subdomain

This is the Differentiator for the business

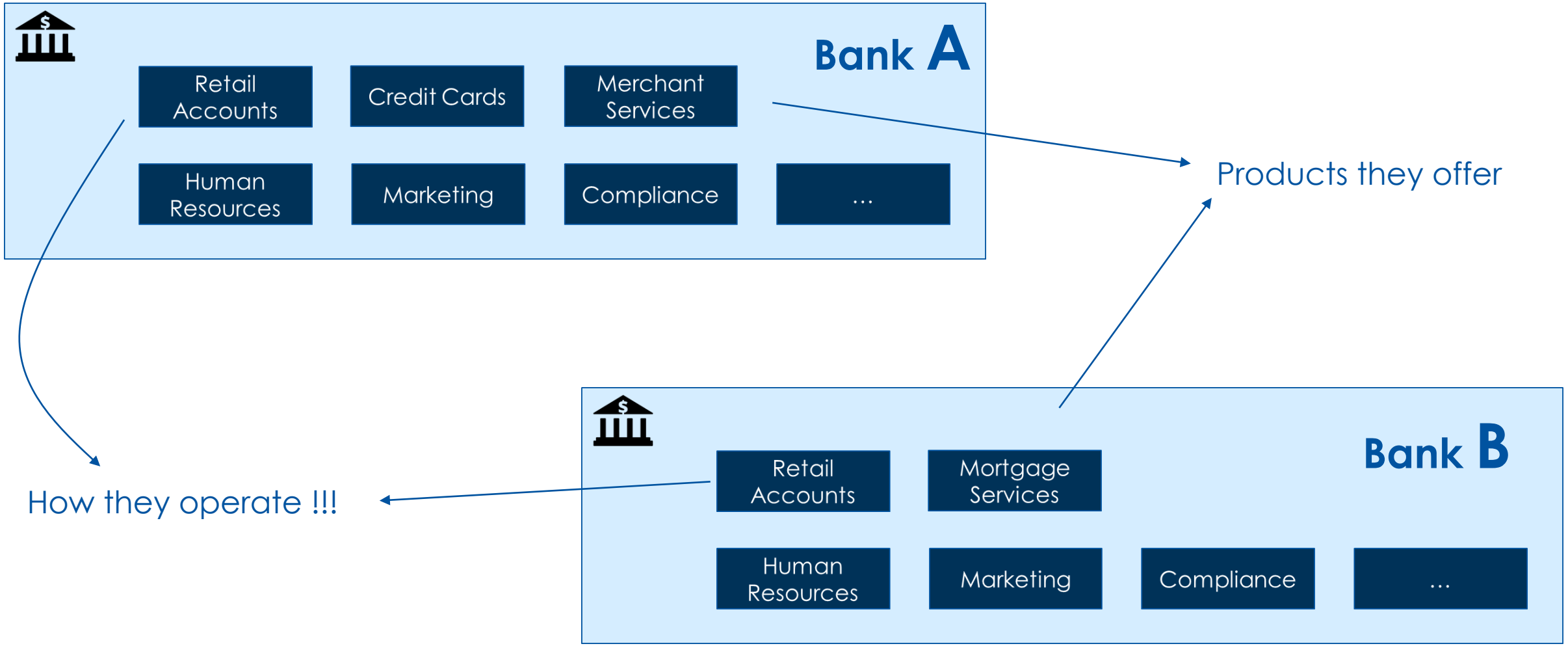
Core

- Secret sauce is in the core subdomain
- Business looks for ways to get competitive advantage
- Fast pace & ever evolving

Manufacturing

Credit Cards

Quick Exercise: What differentiates these banks?



Supporting Subdomain

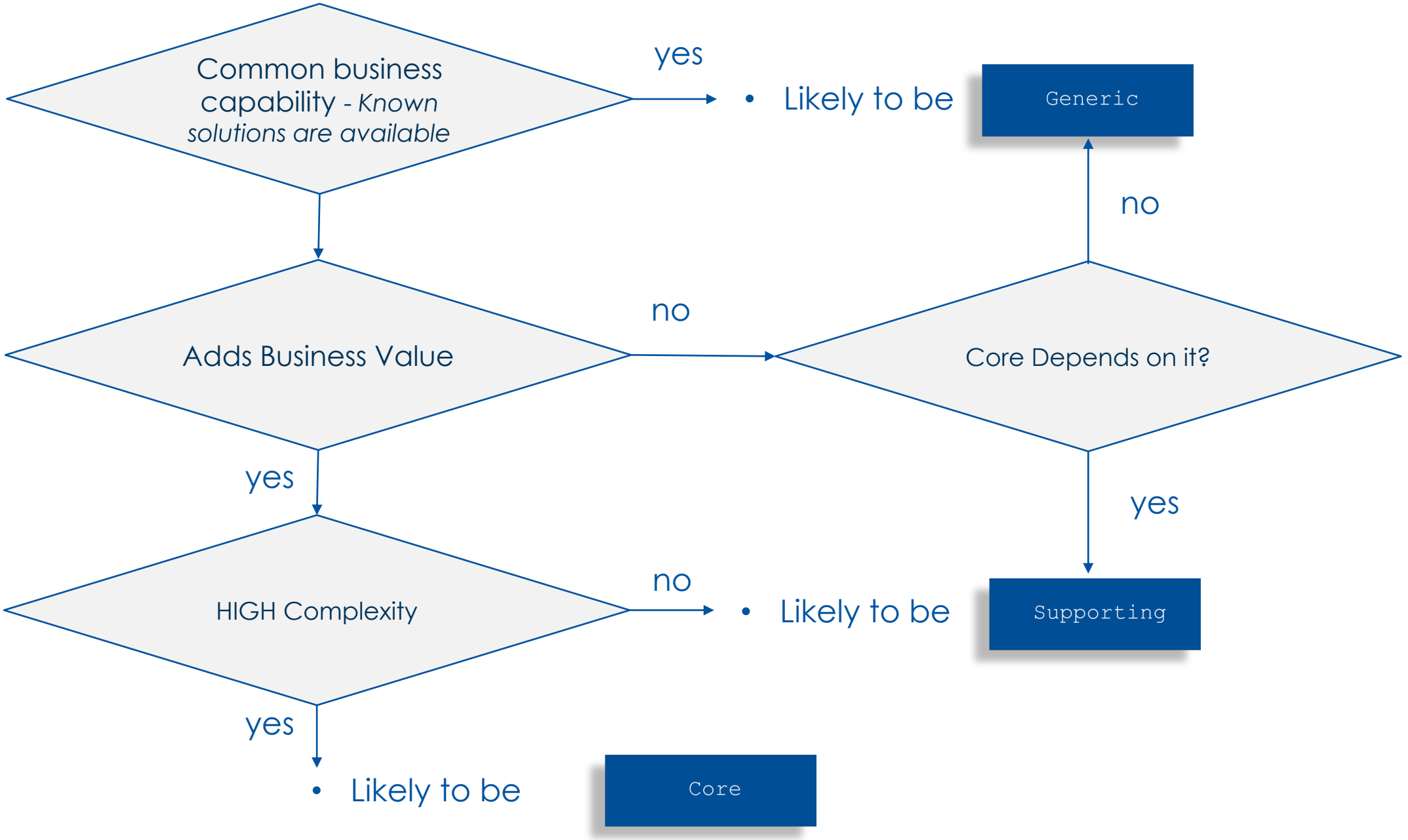
Does not provide business advantage but core depends on it

Supporting

- Well known practices but solutions not readily available
- Relatively simple to build as no complex business logic

Customer Support

Compliance


Common business capability - Known solutions are available
yes
• Likely to be
Generic
no
Adds Business Value
no
Core Depends on it?
yes
yes
HIGH Complexity
no
• Likely to be
Supporting
yes
• Likely to be
Core

Why do we categorize the sub-domains?

- Limited Resource; drives prioritization
- Return on Investments
- Build Vs. Buy decisions

IT Solutions : Buy Vs Build

Generic

- Buy

Supporting

- Outsource
- Customized

Core

- Built by '***A***' teams

Quick Review

- 3 type of Subdomains `Generic` `Core` `Supporting`
- Categorization helps with build vs. buy decisions
- Businesses gets the most ROI by investing `Core`

Business Context

Understanding the meaning of the context

1. What is Context?
2. What is Business Context?
3. Why is it important for IT Teams to understand the context?

1. Should I prefer an account with **High** interest rates

2. Should I prefer an account with **Low** interest rates

What would you tell Jack to do?

1. Should I prefer an account with **High** interest rates

2. Should I prefer an account with **Low** interest rates

This is good if Jack is looking to open a "SAVINGS Account"

1. Should I prefer an account with **High** interest rates

2. Should I prefer an account with **Low** interest rates

This is good if Jack is looking to open a "CREDIT CARD Account"

1. Should I prefer an account with **High** interest rates

2. Should I prefer an account with **Low** interest rates

To make an objective decision you will need additional facts or information on the situation !!!

You need the *Context*

Context

The circumstances or facts that form the setting for a statement, event or idea

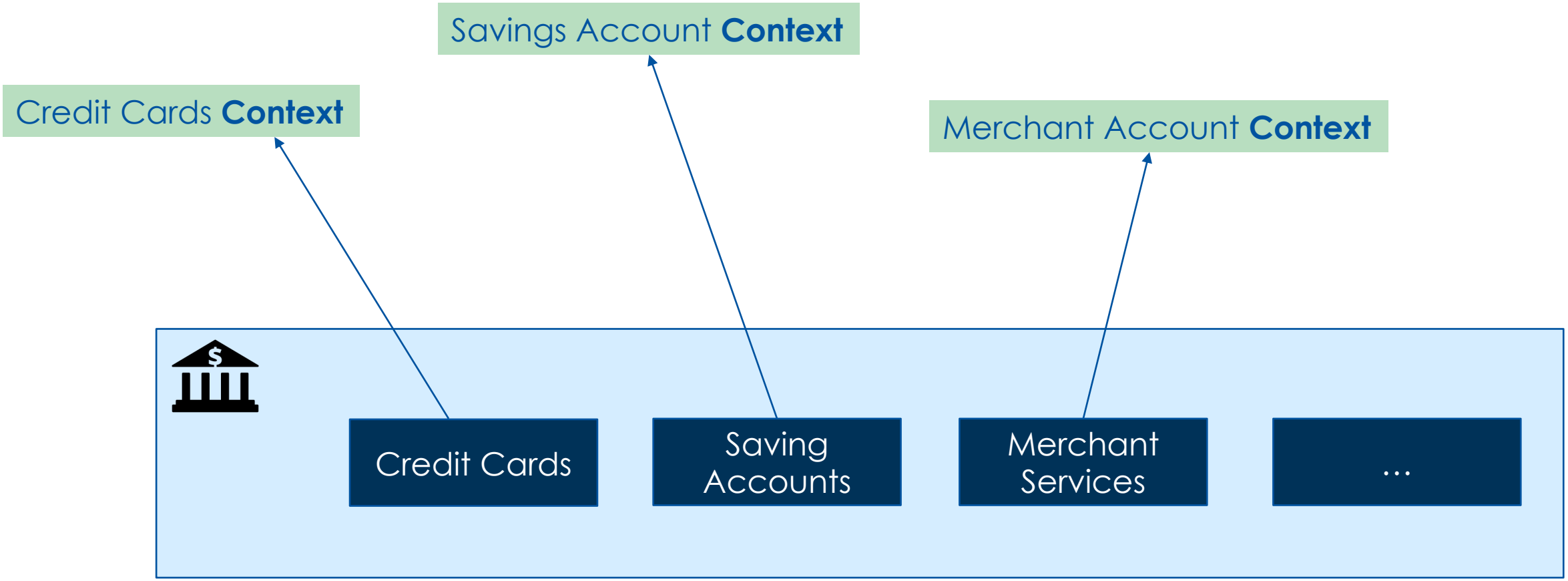
Context: Jack is looking borrow money from the bank.

1. Should I prefer an account with **High** interest rates

2. Should I prefer an account with **Low** interest rates

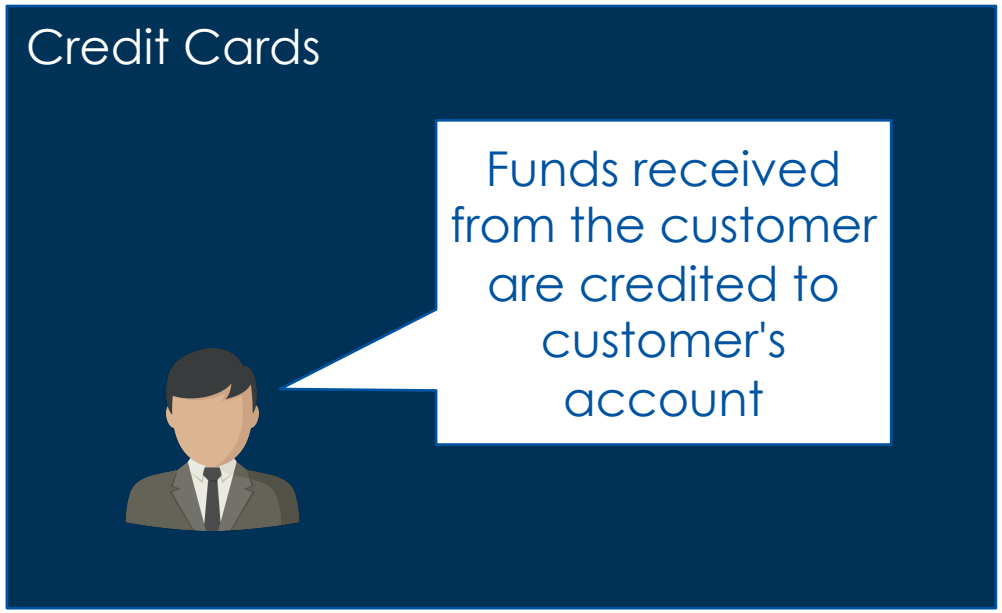
Business Context

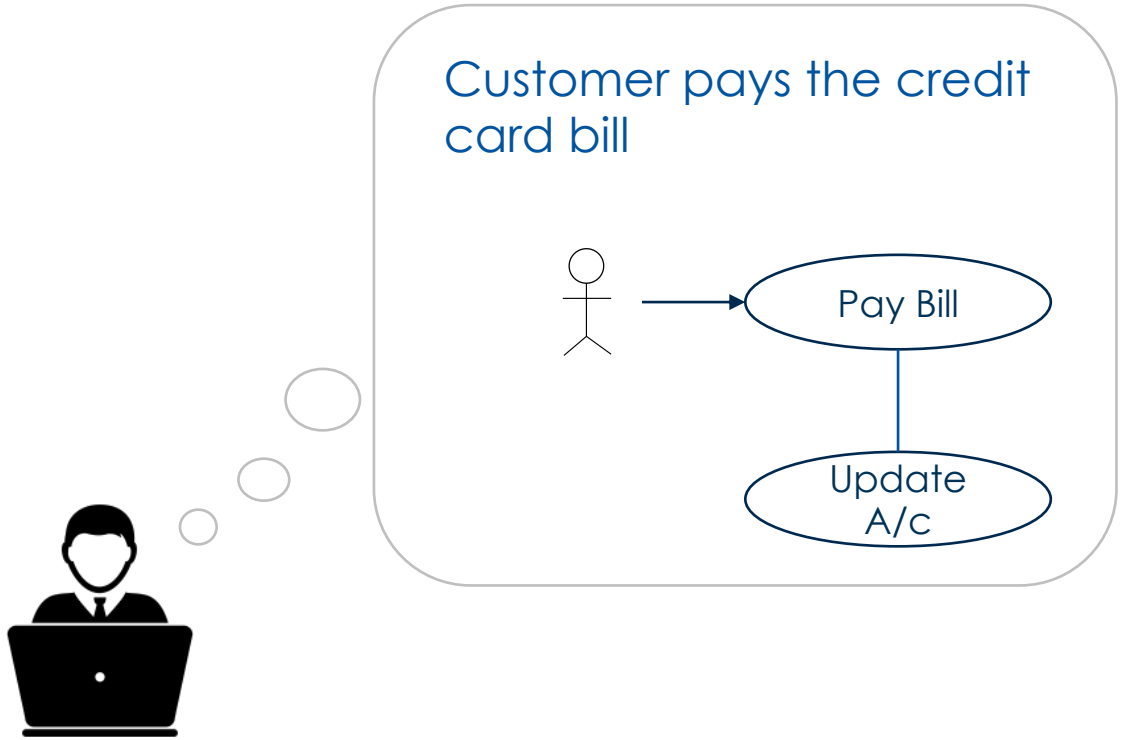
Focus areas for knowledge crunching



Business Context

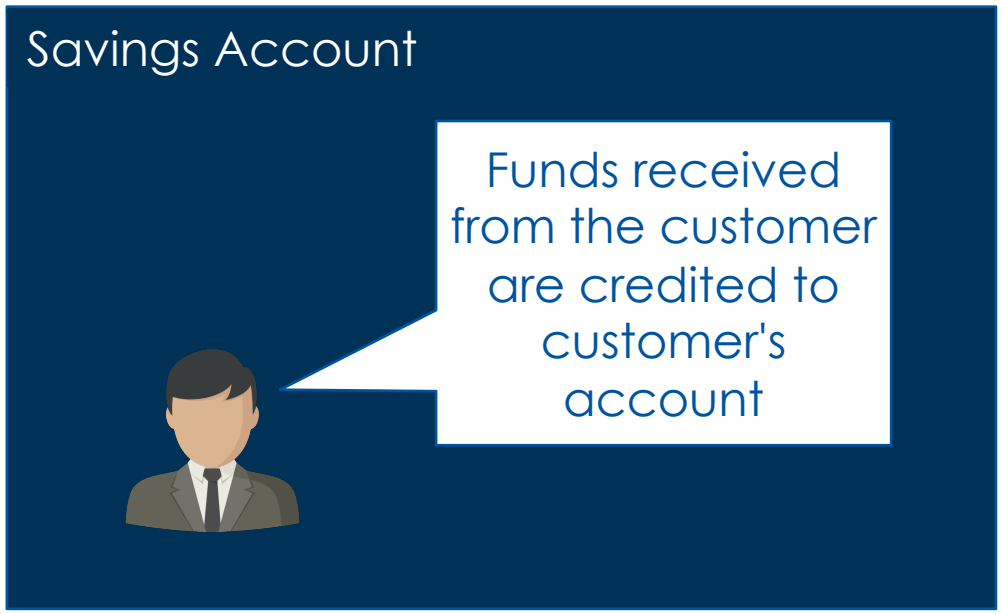
In order to understand the business domain one MUST understand the business context

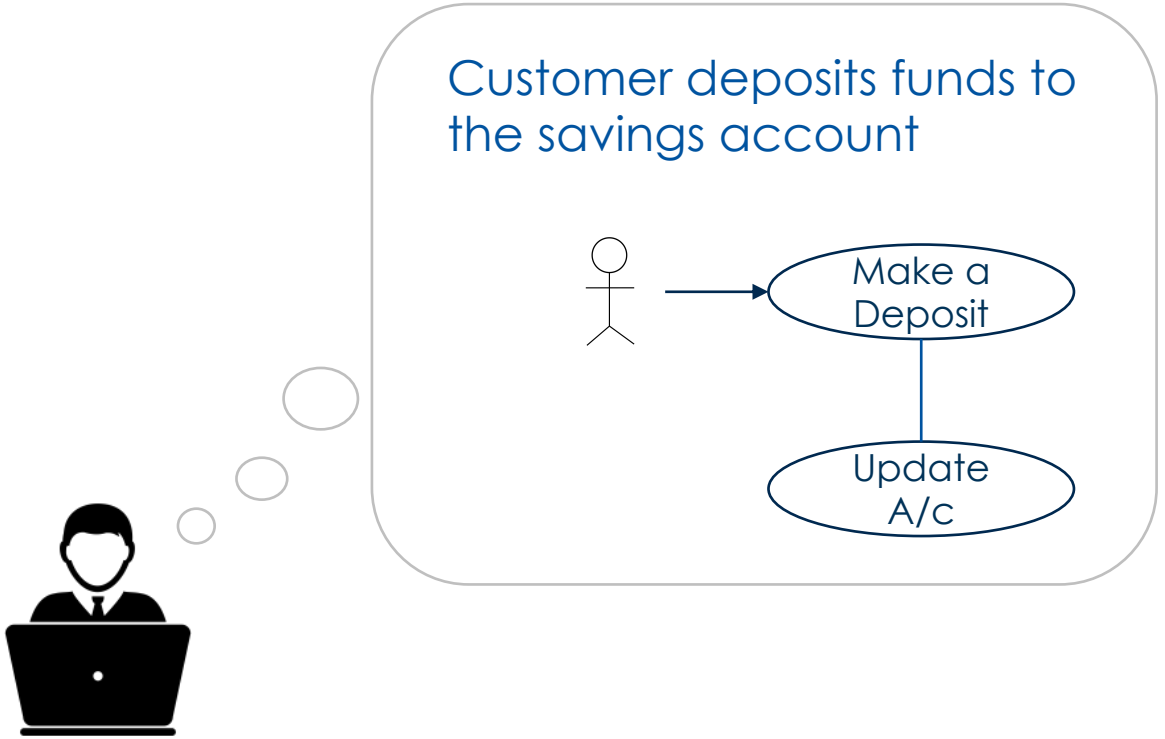




Business Context

In order to understand the business domain one MUST understand the business context





Why so much emphasis on business context?

- Confusion & Misinterpretation

- Leading to Misrepresentation of the domain model(s)

Quick Review

- Context is the circumstances or facts that form the setting for statement, event or an idea

- For understanding the Business Domain it is a MUST that IT teams understand the business context

Business Language

IT team MUST learn the language of business

Business Language

Linguistic Challenges

Introduction to Ubiquitous Language

Tuna

Japanese = Large Fish

Spanish = Cactus

Pasta

Italian = Noodle

Polish = Toothpaste

Kiss

Swedish = Pee ☺

Same terms used in different region may lead to confusion

Business Language

Each industry | profession has its own lingo

Business Language

There are specializations within an industry

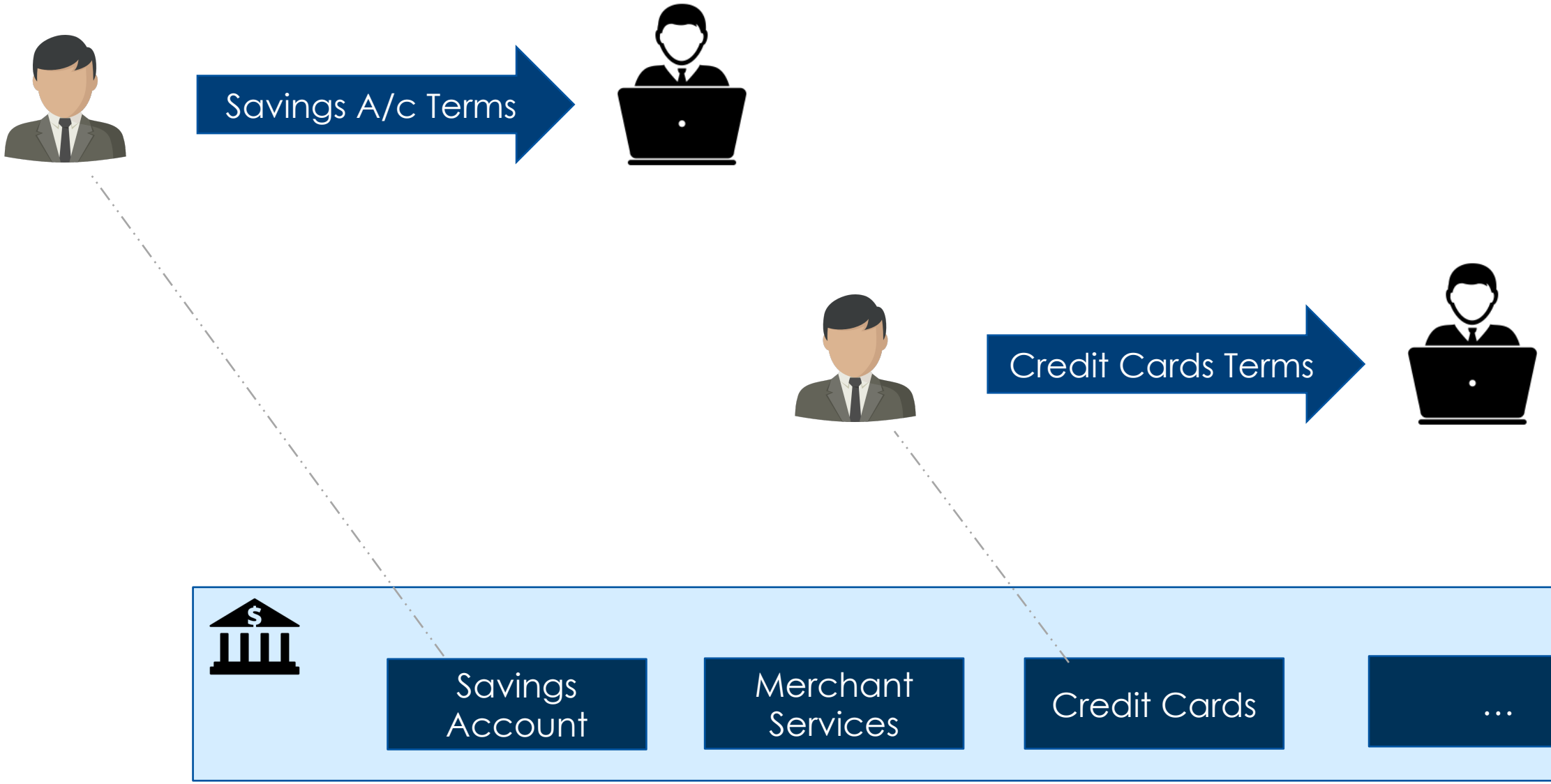
- Auditor
- Management accountant
- Project accountant
- Certified Public Accountant
- Forensic accountant
- .....

Business Language

- Teams within enterprise have their own lingo

- Experts use the team's language in all their communication

Business Language



Understanding the domain

To gain knowledge of the domain one MUST understand the language used by the domain experts

Linguistic Challenges

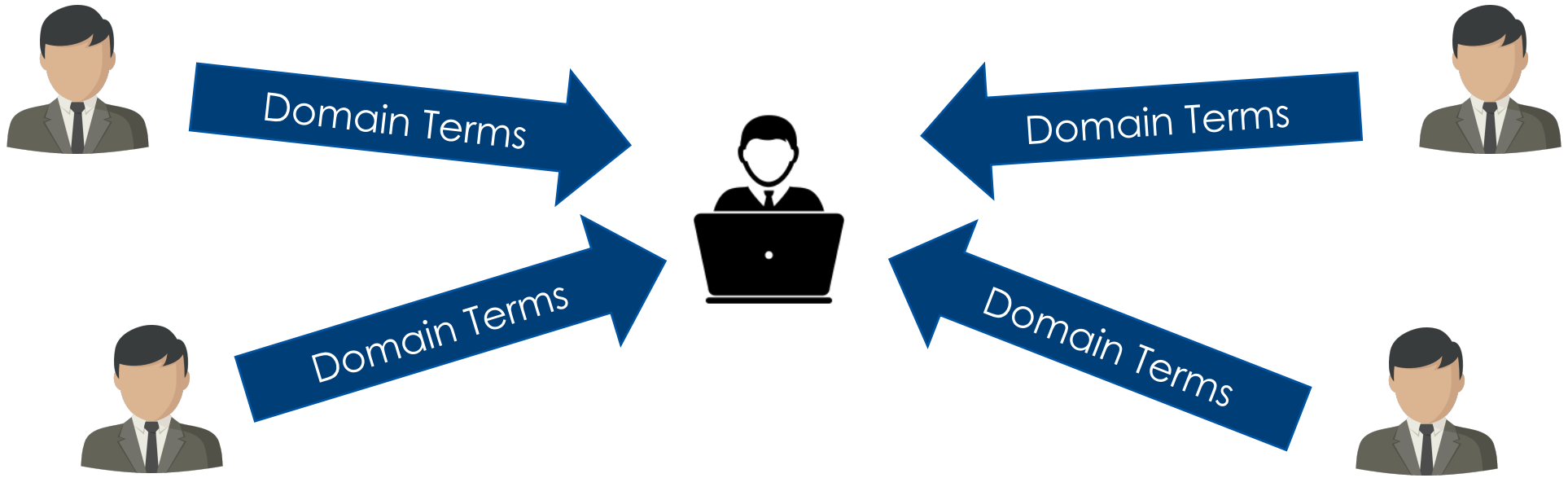
1 Multiple business languages across the enterprise

- Credit
- Deposit
- Withdraw
- Debit

- Credit
- Payment
- Merchant
- Cash advance

Linguistic Challenges

1 Multiple business languages across the enterprise



For building complex systems IT Teams MUST learn multiple Business languages used by the Experts in the context of different domains

Linguistic Challenges

2

Conflicting meaning of terms

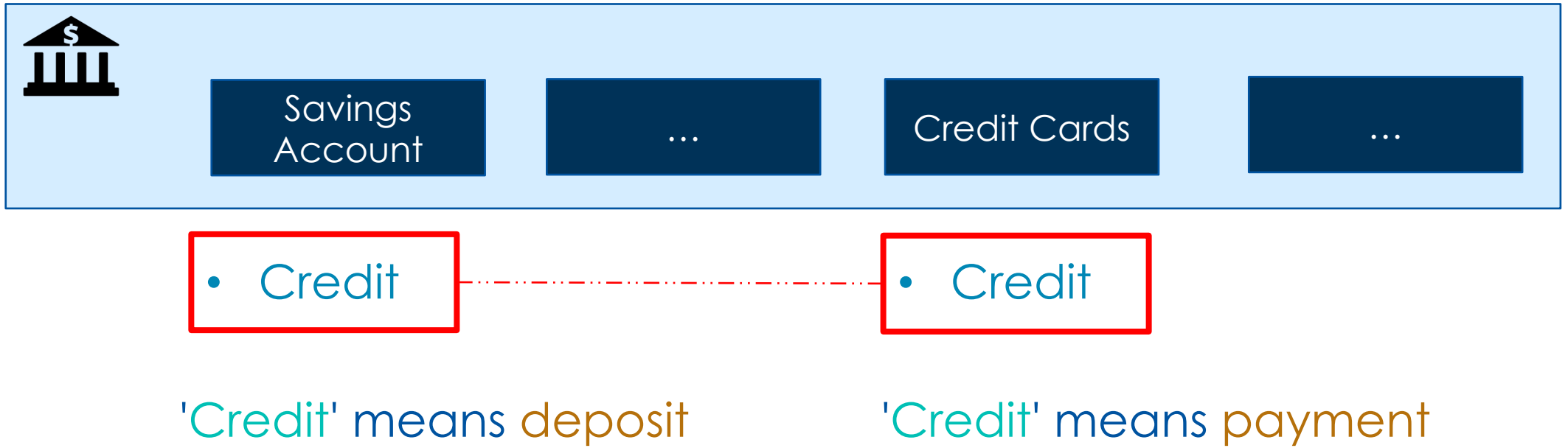
- Credit
- Deposit
- Withdraw
- Debit

- Credit
- Payment
- Merchant
- Cash advance

Linguistic Challenges

2

Conflicting meaning of terms



Different meaning depending on the context is a cause of confusion!!

Linguistic Challenges

3 Technology teams tend to translate business terms to IT lingo

Domain Terms

Technology Terms

- getters / setters
- create / delete
- Object
- DB
- …

Linguistic Challenges

3

Technology teams tend to translate business terms to IT lingo

We check customer's credit history

I open a Savings A/c for customer

Deposit the initial $ to account

Customer deposits funds to the savings account

- Call getCreditHistory()
- CreateAccountInDB()
- SetAmountAcObject()

Linguistic Challenges

3 Technology teams tend to translate business terms to IT lingo

We check customer's credit history

I open a Savings A/c for customer

Deposit the initial $ to account

Translation

Customer deposits funds to the savings account

- Call getCreditHistory()
- CreateAccountInDB()
- AddToAccountObject()

Software doesn't reflect domain's terms; loss of meaning & confusion due to back & forth translation between IT <> Domain

How are these linguistic challenges addressed?

Domain Driven Design suggests setting up a common language within each business context; this language is used by ALL stakeholders

Ubiquitous Language

Quick Review

- IT Teams MUST learn business language
- Confusion caused by translation between domain & IT
- DDD suggests creation of common language that MUST be used by all stakeholders

Ubiquitous Language

DDD Strategic pattern to address Linguistic Challenges

Characteristics of UL

Building & Managing the UL

Where to use UL

Strategic Pattern: Ubiquitous Language

DDD suggests setting up a common language (set of terms) within each business context

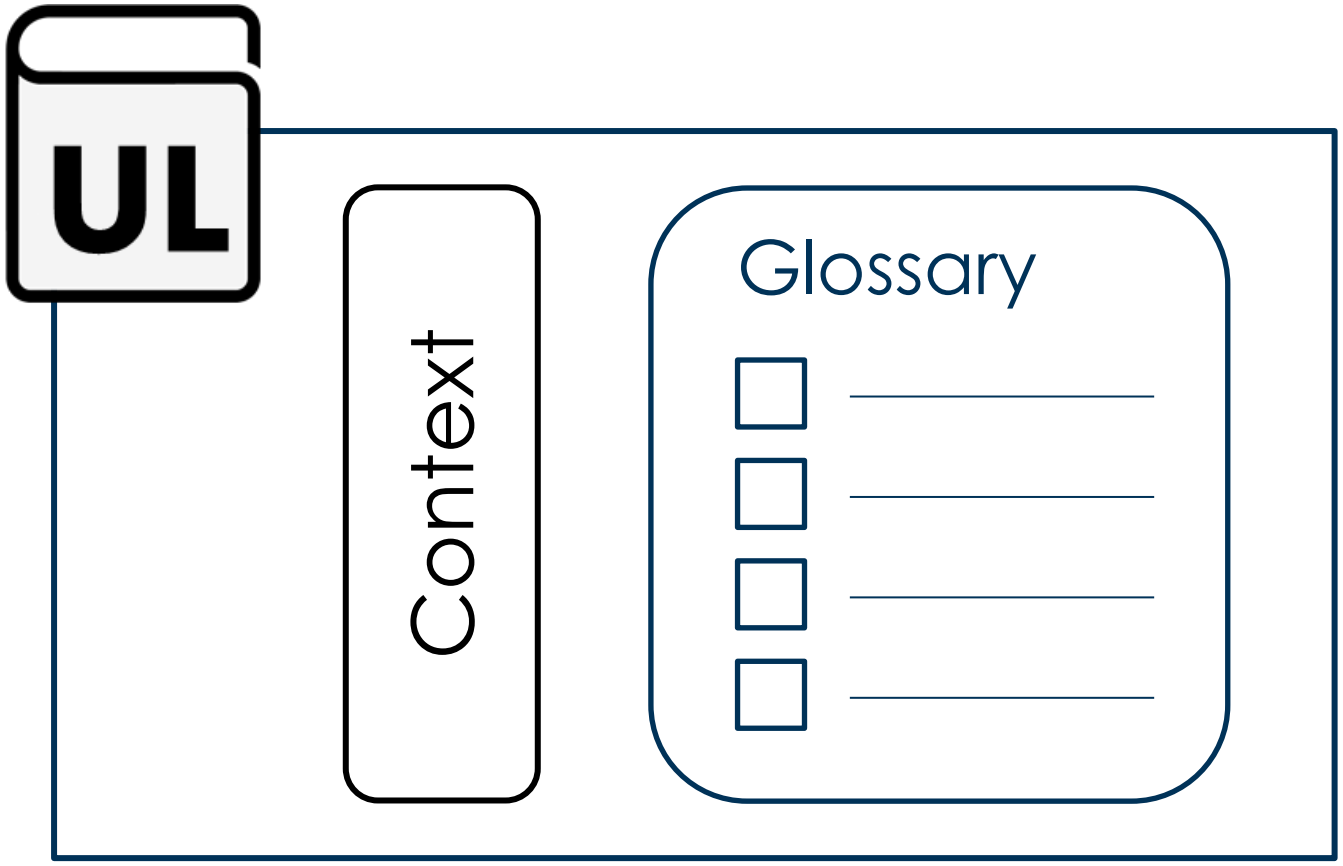
Ubiquitous = *Constantly encountered or seen*

Linguistic Challenges

Addressed by way of UL

Characteristics of UL

Dialect defined by the vocabulary

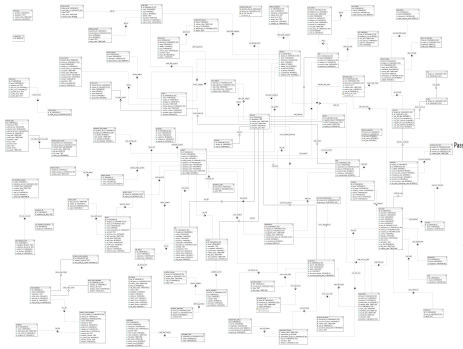


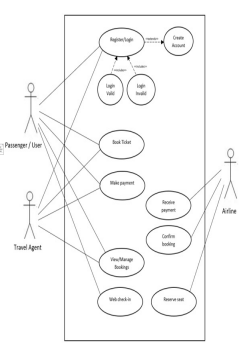
Terms & Acronyms

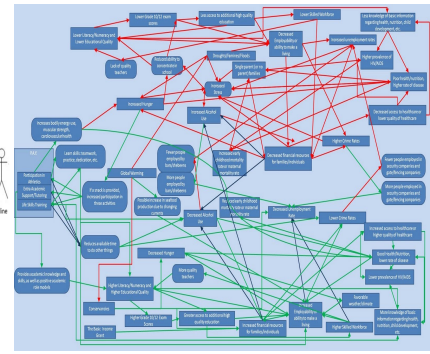
Examples

Links to relevant assets

Common Business Dictionary







Business Dictionary

☐ _____

☐ _____

☐ _____

☐ _____

IF product='auto' THEN

It means …

ELSE IF product='home' THEN

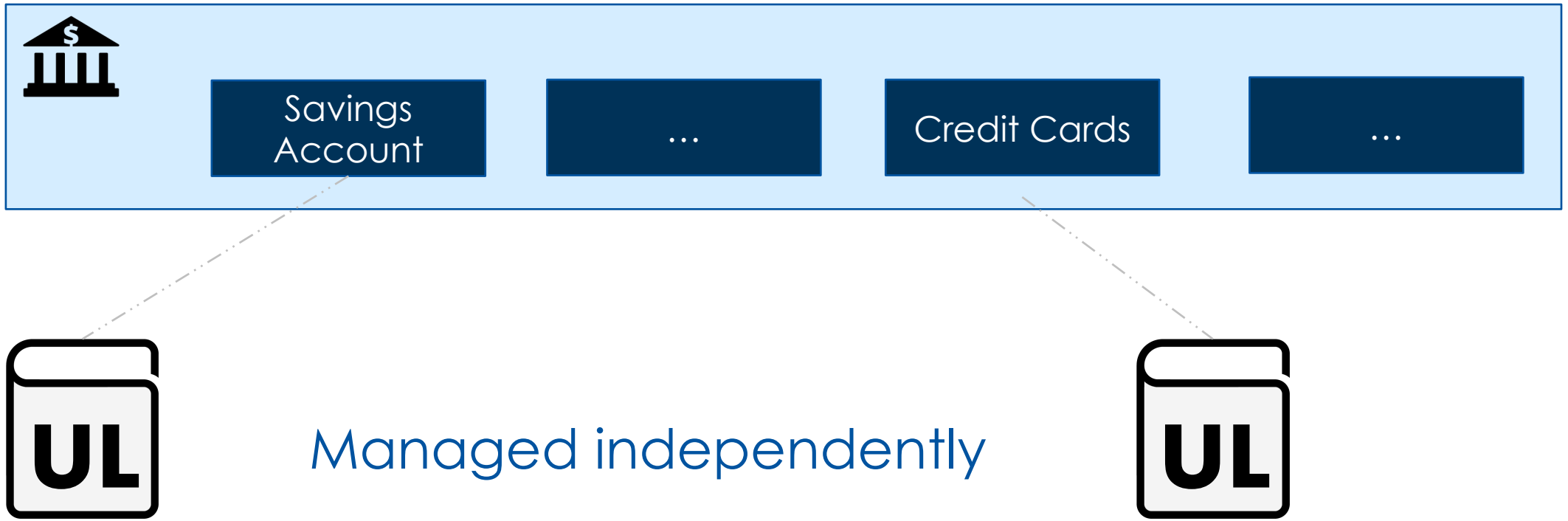
It means …

ELSE …

Managed Centrally by "Business Analyst" in an Excel sheet !!!

Ubiquitous Language

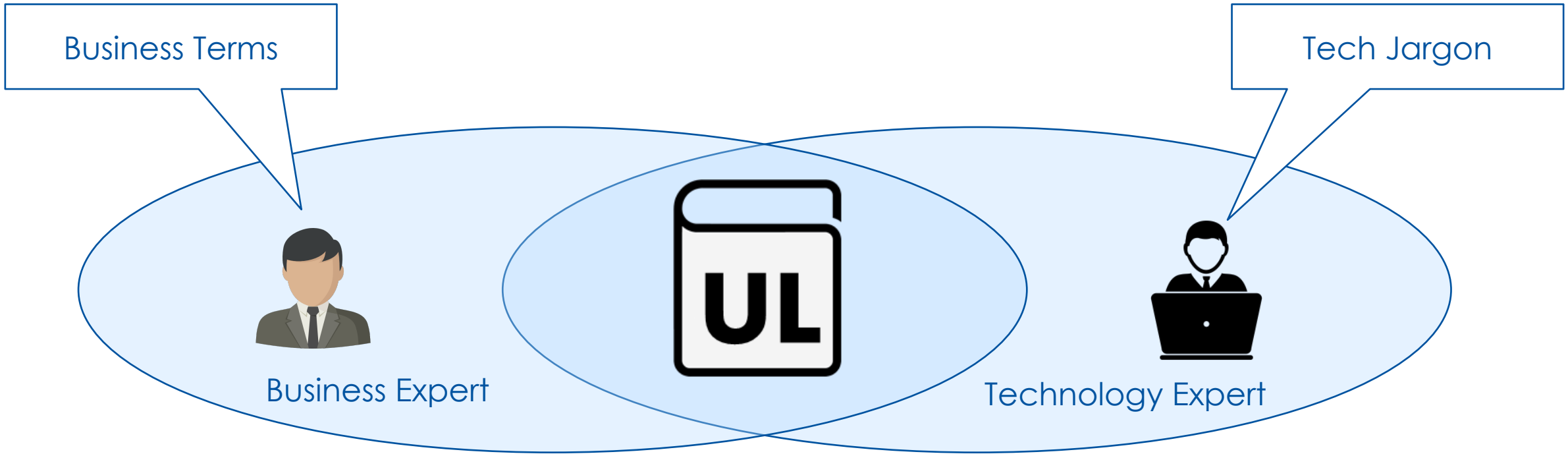
Multiple UL in an organization



Characteristics of UL

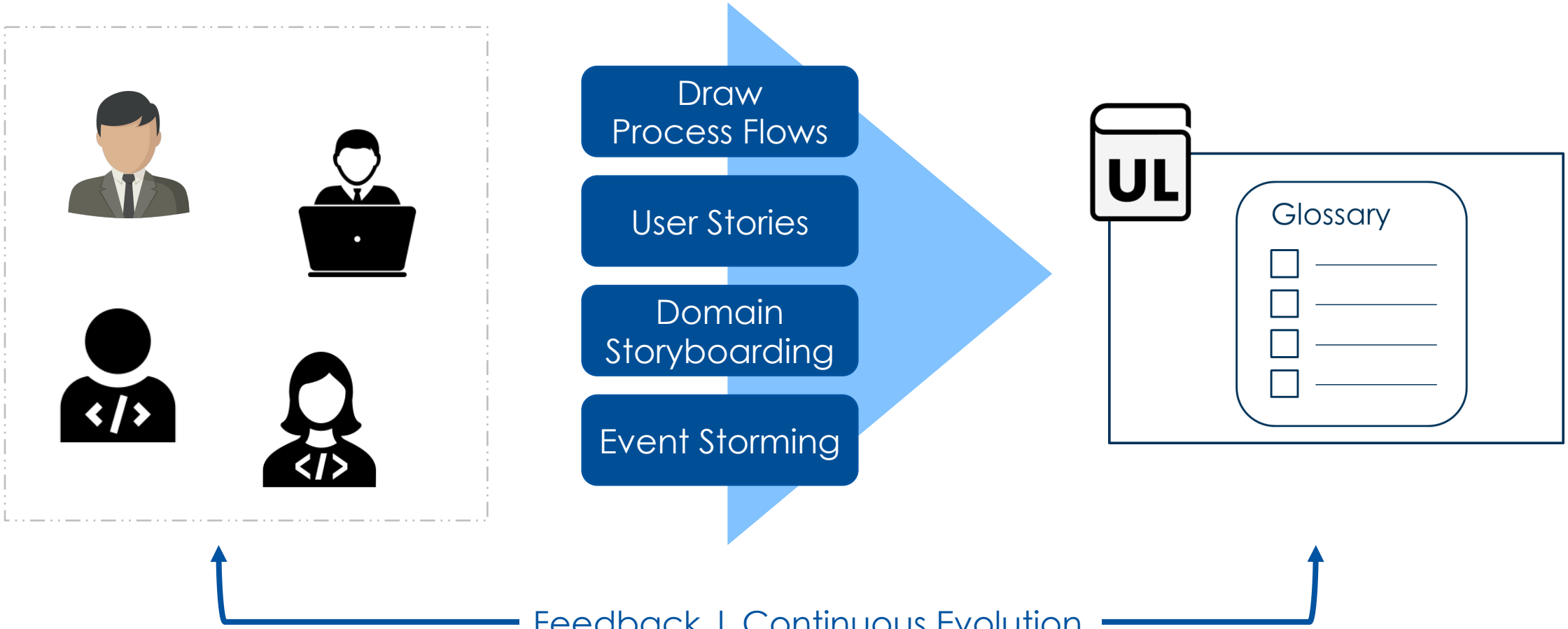
Not the business language imposed by experts

Not the language used in the industry



Characteristics of UL

Developed collaboratively by Business experts & IT experts



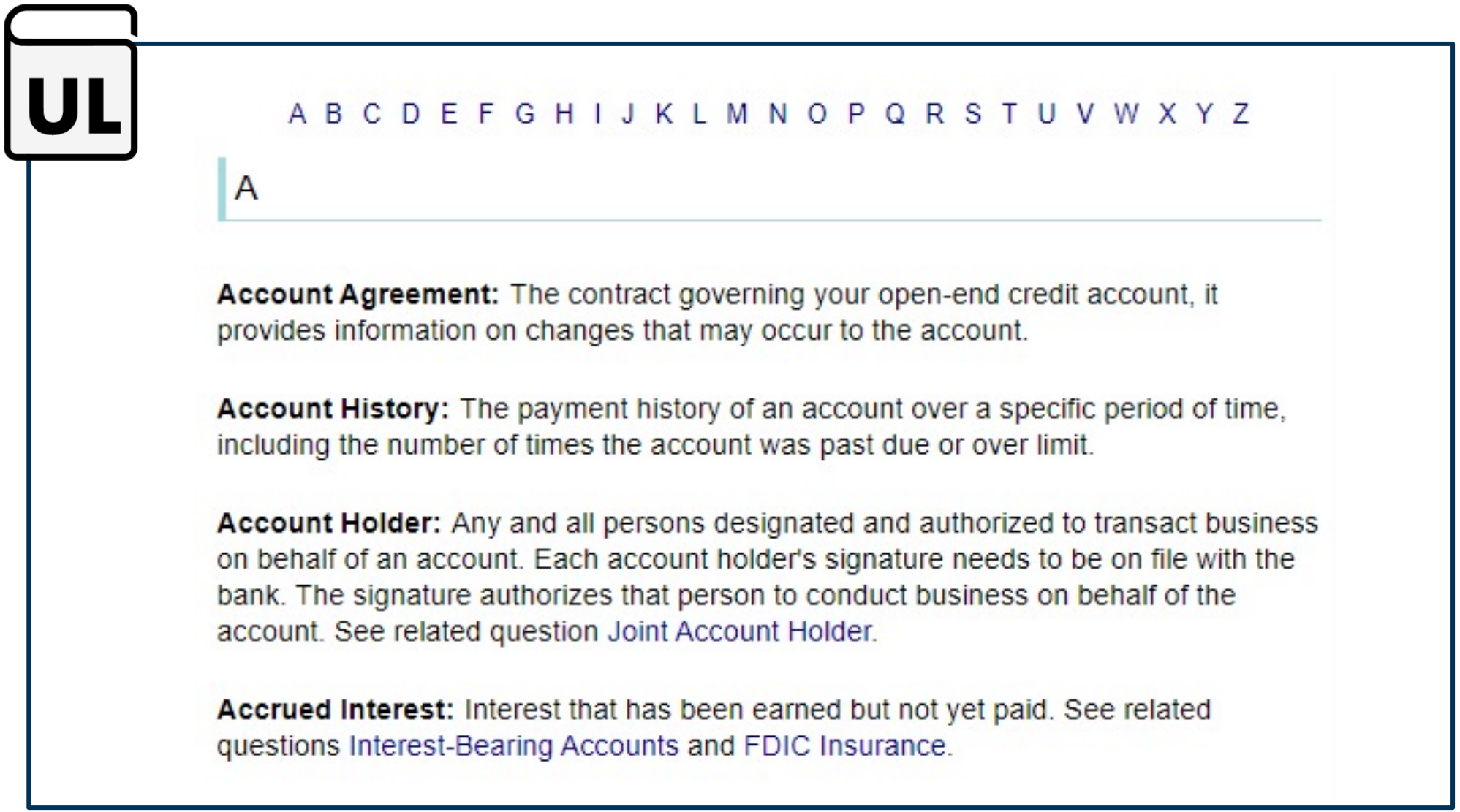
Are there any tools for UL?

MUST be easily accessible to all team members

Collaboration wiki tool for knowledge sharing

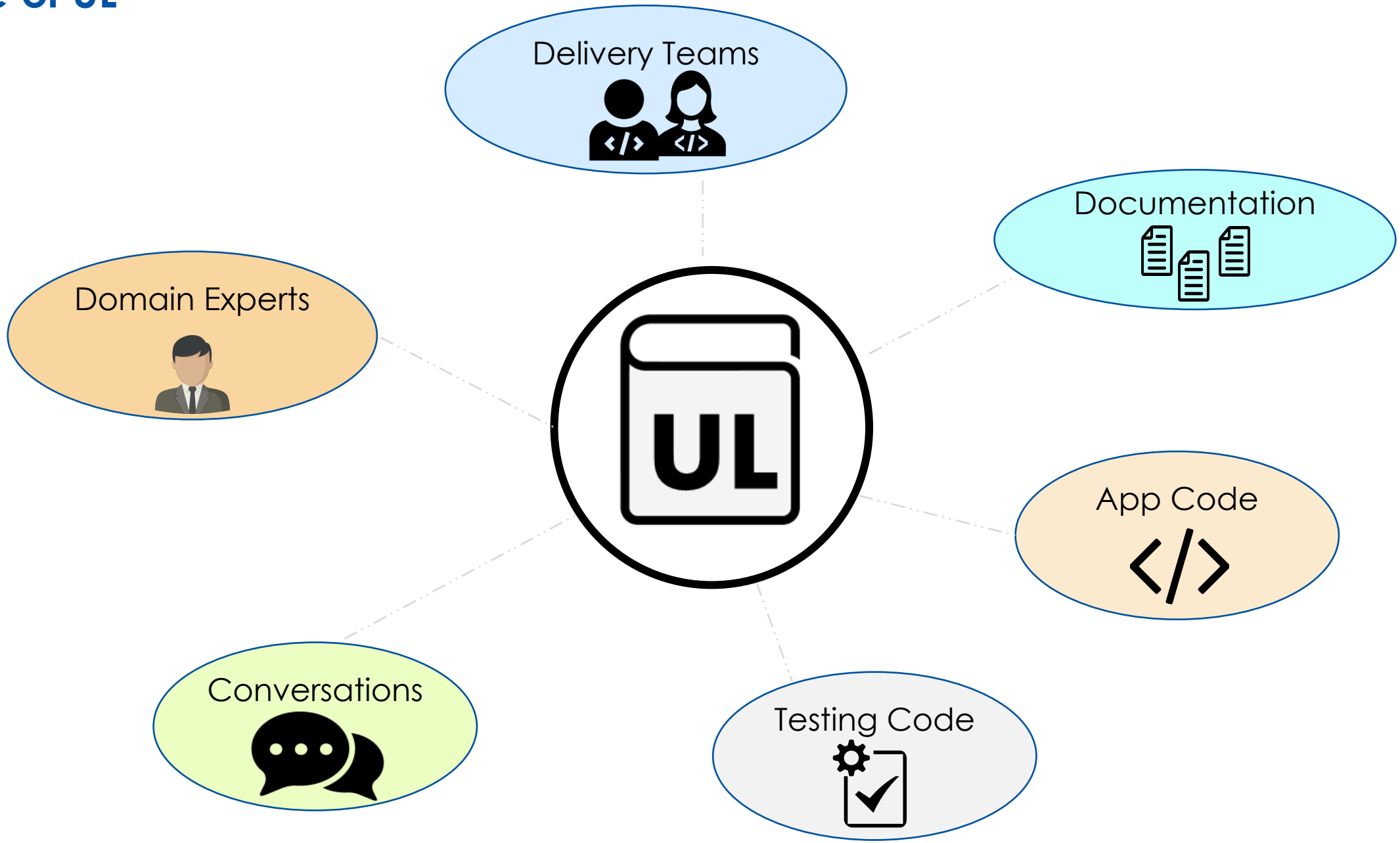
Salesforce collaboration tools

Example: Retail Accounts



https://www.helpwithmybank.gov/dictionary/index-dictionary.html

Use of UL


Delivery Teams
Documentation
Domain Experts
UL
App Code
Conversations
Testing Code

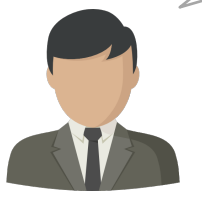
Address Linguistic Challenges - Translation

Technology teams tend to translate business terms to IT lingo

We check customer's credit history

I open a Savings A/c for customer

Deposit the initial $ to account

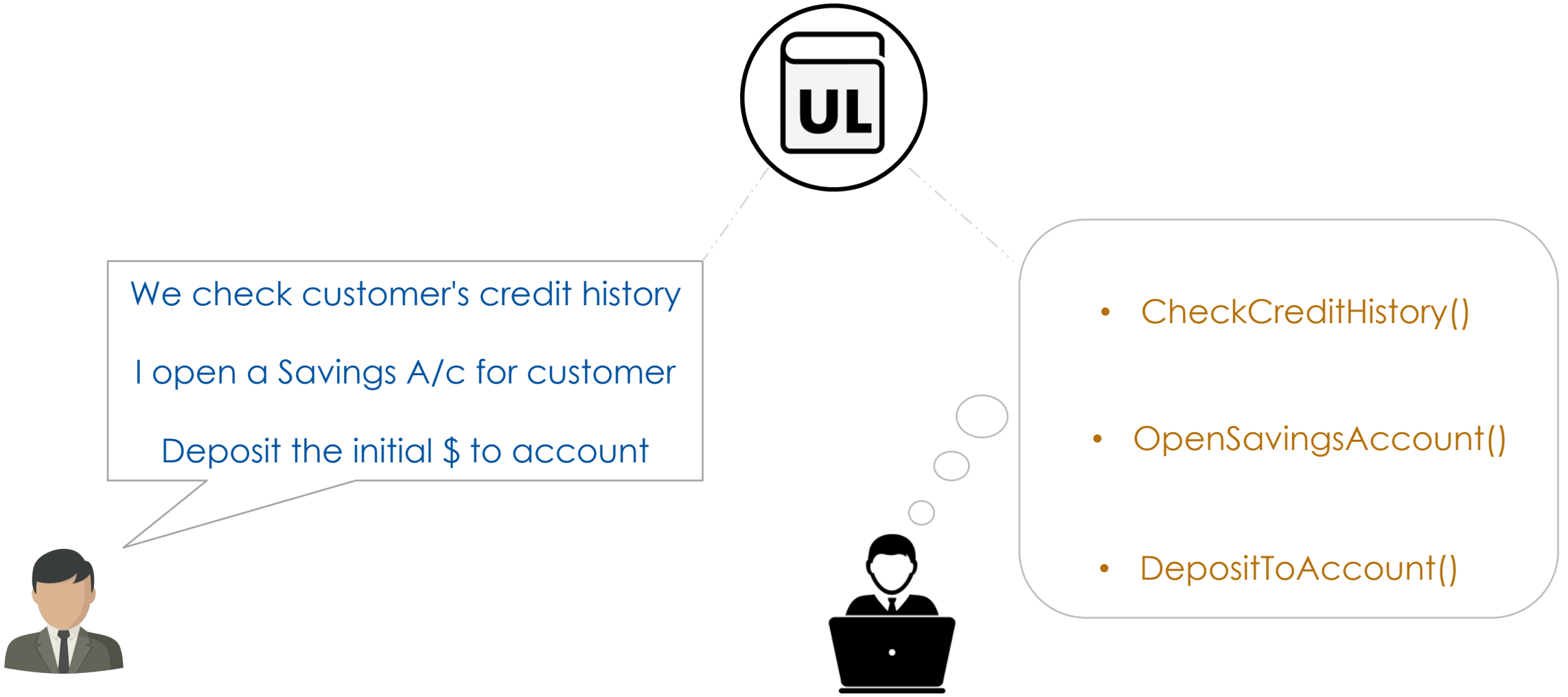


Translation

Customer deposits funds to the savings account

- Call getCreditHistory()
- CreateAccountInDB()
- AddToAccountObject()

UL Addresses the Linguistic Challenges



UL helps in identifying overlapping contexts

> Ubiquitous Language helps in breaking the business context into smaller parts; in DDD these smaller parts are referred to as the *Bounded Context*

Bounded Context

Quick Review

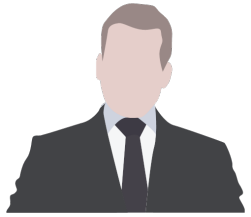
- UL evolves over a period
- UL may be managed on any knowledge collaboration platform
- UL MUST be used in all form of communication & code

Exercise : ACME's UL Terms

Review of interview transcripts of two critical ACME stakeholders

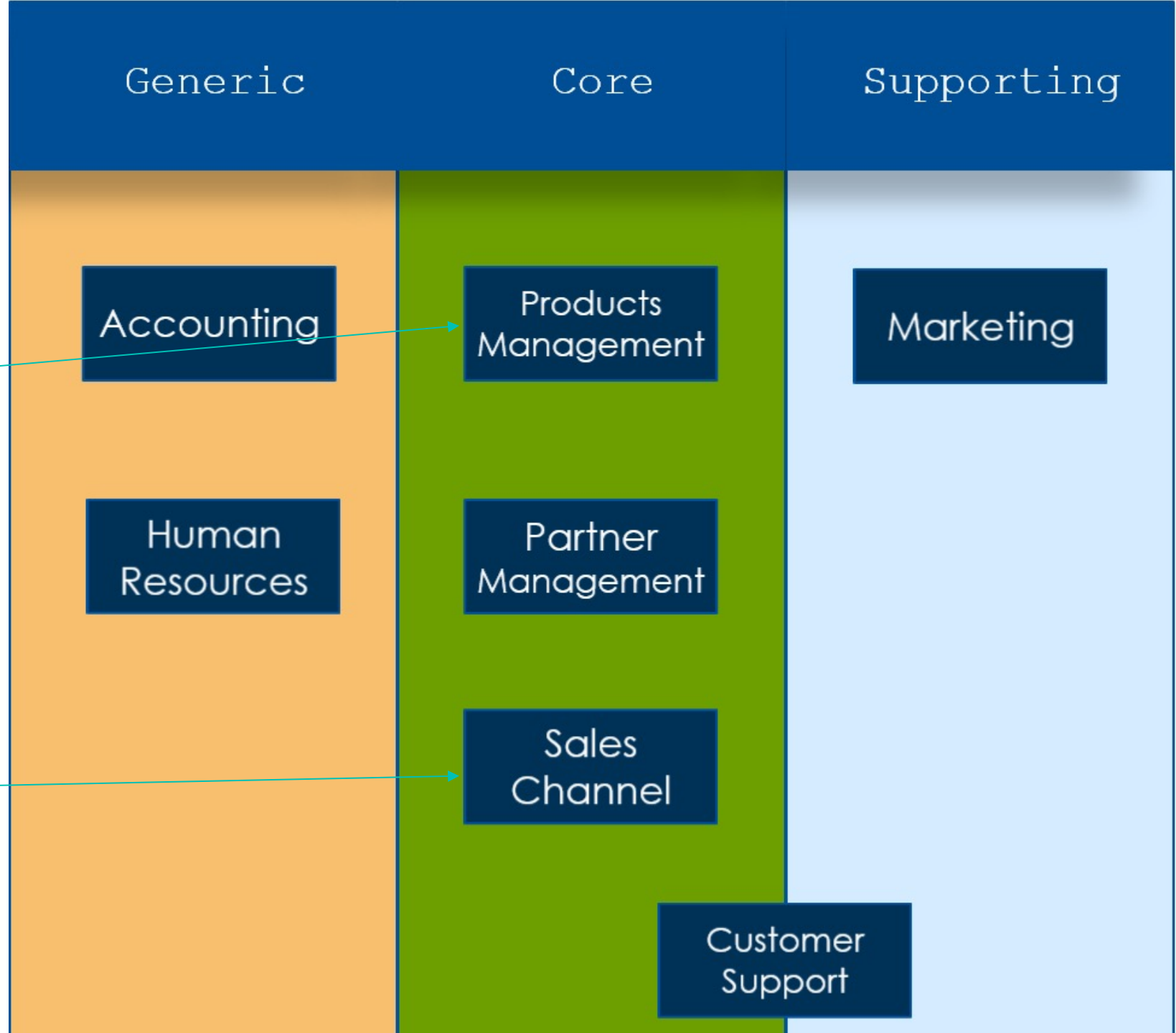
1. Identify the UL Terms - Sales Team
2. Identify the UL Terms - Product Management

Meet the Experts

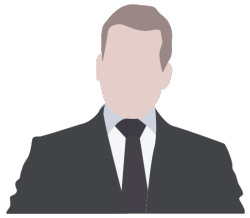


Paul, Product Manager

John, Travel Advisor



Business Experts



Paul, Product Manager

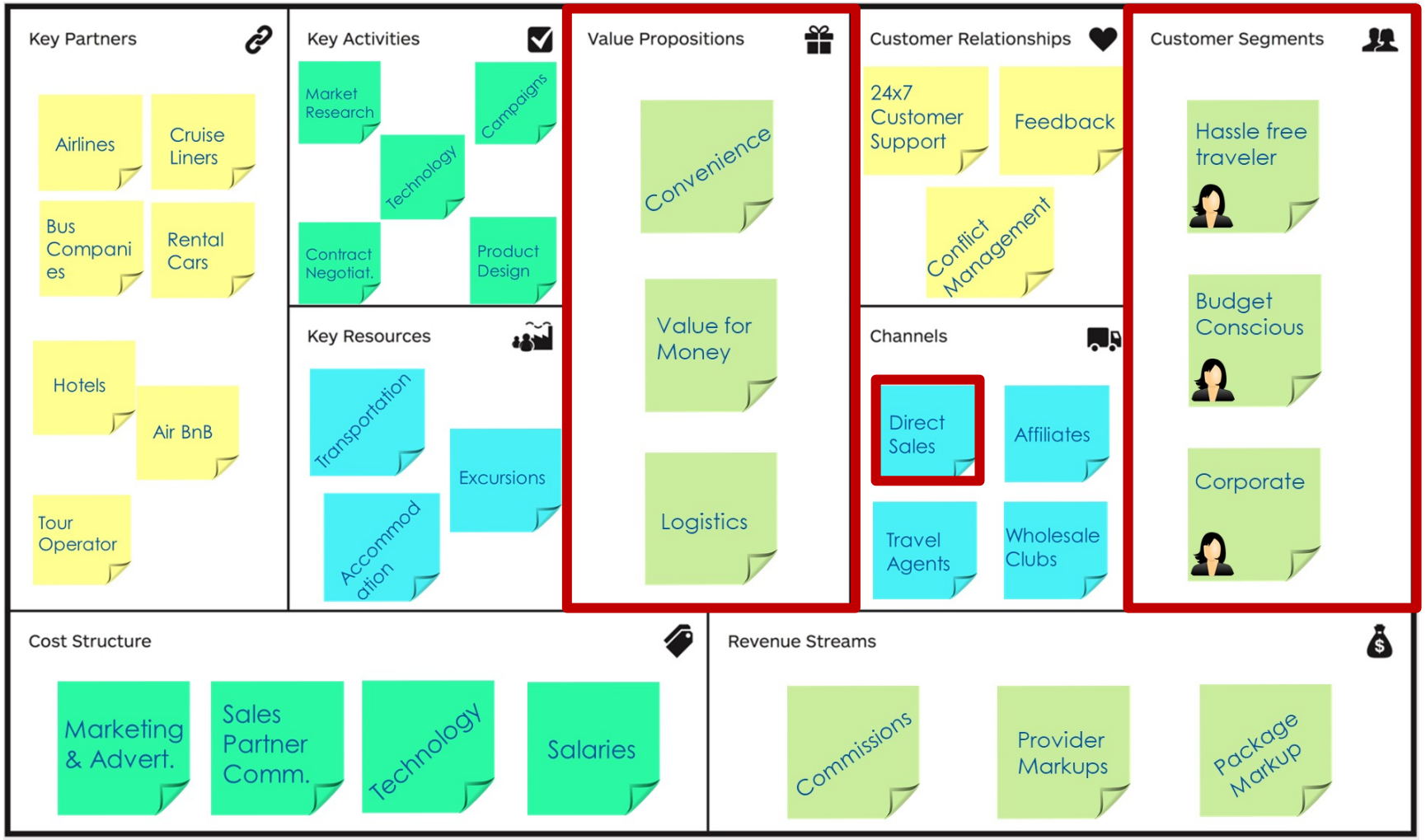
- Specializes in designing the vacation packages
- Manages the relationship with the partners

John, Travel Advisor

- Helps customers select the vacation package
- Understand the customer's needs

John, Travel Advisor

- Understand the customer's needs
- Helps customers select the vacation package



John, Travel Advisor

Identify the UL terms

John, Travel Advisor

IT Lead

We are in the business of selling vacation packages. The sale process starts with a Customer calling us. Based on Customer's desires we select the packages and describe it. If customer shows interest we start a proposal for the selected package.

Packages have a suggested retail price but our product team also puts out offers that we can apply to the packages. These offers are essentially the discounts based on various criteria.

Once the customer commits to the proposal, we gather the Pax details i.e., passengers details. All the information is gathered into a purchase order and then we get the Payment information submit the proposal for reservation and if everything goes fine we receive the Booking confirmation.

John, Travel Advisor

IT Lead

We are in the business of selling vacation packages. The sale process starts with a Customer calling us. Based on Customer's desires we select the packages and describe it. If customer shows interest we start a proposal for the selected package.

Packages have a suggested retail price but our product team also puts out offers that we can apply to the packages. These offers are essentially the discounts based on various criteria.

Once the customer commits to the proposal, we gather the Pax details i.e., passengers details. All the information is gathered into a purchase order and then we get the Payment information submit the proposal for reservation and if everything goes fine we receive the Booking confirmation.

Vacation Package

Customer

Proposal

Offers

Pax or Passenger

Purchase Order

Payment Information

Booking Confirmation

John, Travel Advisor

Package Pricing

Lump Sump Pricing

Satisfaction Index

All-inclusive

Partners

Itinerary

(PNR) Passenger Name Record

(BAR) Best Available Rate

Market Price

(PO) Purchase Order

Rack Rate

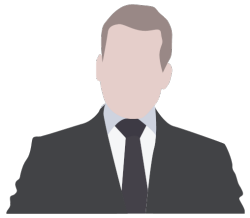
Blackout Dates

Vacation Package

Credit Card Hold

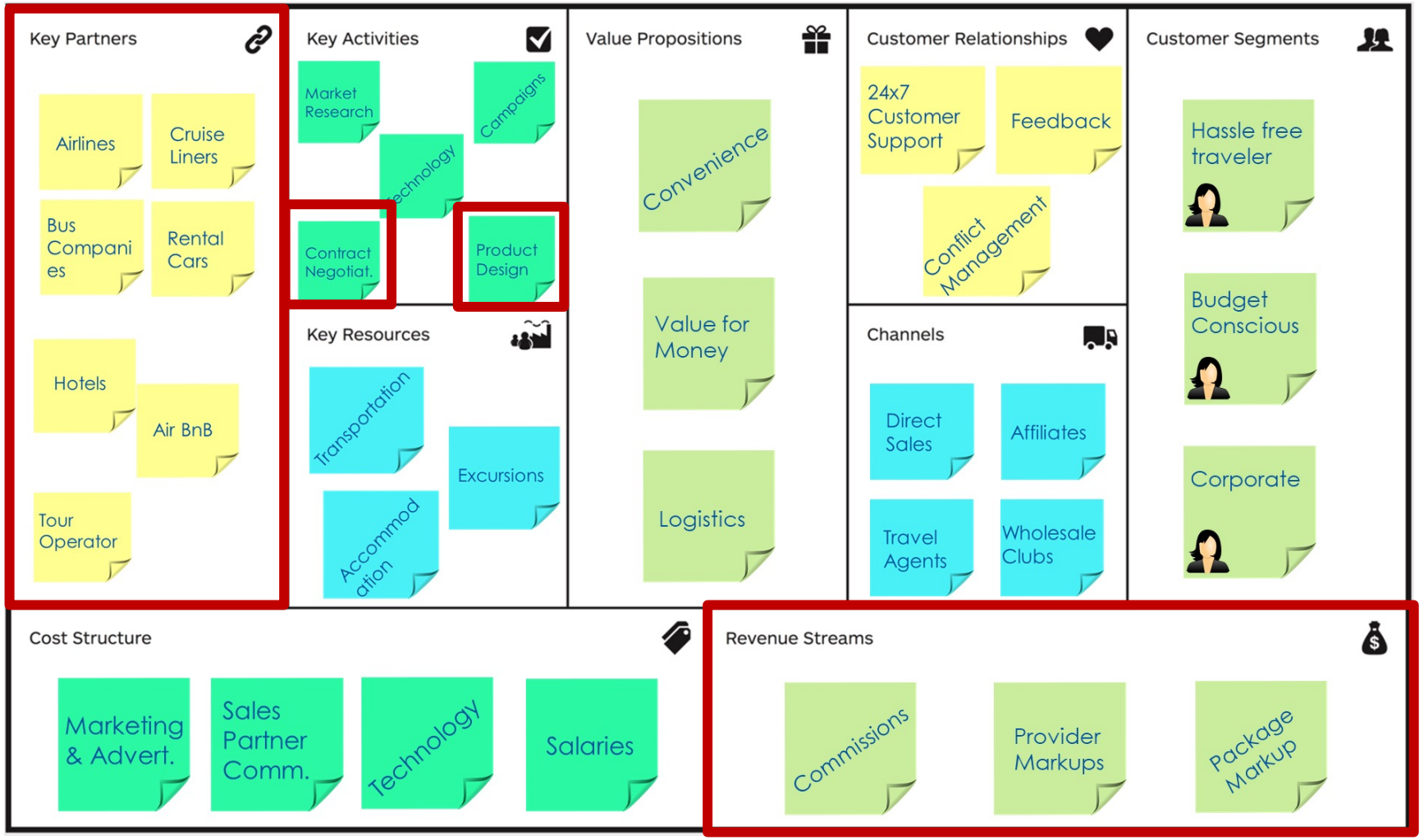
(SLA) Service Level Agreement

… …



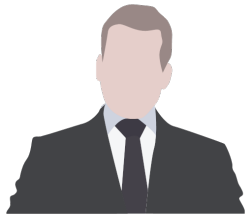
Paul, Product Manager

- Specializes in designing the ACME products
- Manages the relationship with the partners



Paul, Product Manager

Identify the UL terms



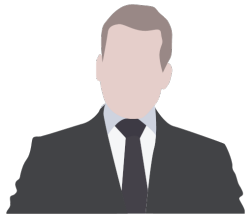
Paul, Product Manager

IT Lead

I am responsible for the product design and provider relationships. These products are what customers buy from Acme.

Based on the market research I pick up the parts of the product, sometime we refer to these products as bundle. There are certain markup guidelines that I need to follow in order to make the product profitable. Also I need to take into account the seasonality.

Correct pricing of the bundle requires careful negotiations with the providers. Some providers such as airlines & Hotels offer us bulk prices which are below the Market Prices. Some providers prefer to work with us on commissions. We sign contracts with providers that lists the commission structure as well as any penalties and the terms.



Paul, Product Manager

IT Lead

I am responsible for the product design and provider relationships. These products are what customers buy from Acme.

Based on the market research I pick up the parts of the product, sometime we refer to these products as bundle. There are certain markup guidelines that I need to follow in order to make the product profitable. Also I need to take into account the seasonality.

Correct pricing of the bundle requires careful negotiations with the providers. Some providers such as airlines & Hotels offer us bulk prices which are below the Market Prices. Some providers prefer to work with us on commissions. We sign contracts with providers that lists the commission structure as well as any penalties and the terms.

Product a.k.a. Bundle

Provider

Customer

Markup

Seasonality

Bulk Prices

Market Price

Commissions

Paul, Product Manager

- Product
- Bundle
- Terms
- Satisfaction Index
- Provider
- Markup
- Pricing
- Catalog
- Seasonality
- Bulk Prices
- Commission Structure
- Contract
- Market Price
- Commissions
- Penalties
- … …

Sales Channel

John, Travel Advisor

Products Management

Paul, Product Manager

Refers to similar concept with DIFFERENT terms

Vacation Package

Product a.k.a. Bundle

Refers to end customer with SAME term

Proposal

Customer

Customer

Provider

Pax or Passenger

Offers

Seasonality

Markup

Payment Information

Purchase Order

No other terms overlap **so far** !!!

Market Price

Bulk Prices

Booking Confirmation

Commissions

Quick Review

- UL evolves over a period

Bounded Context

Strategic pattern in DDD

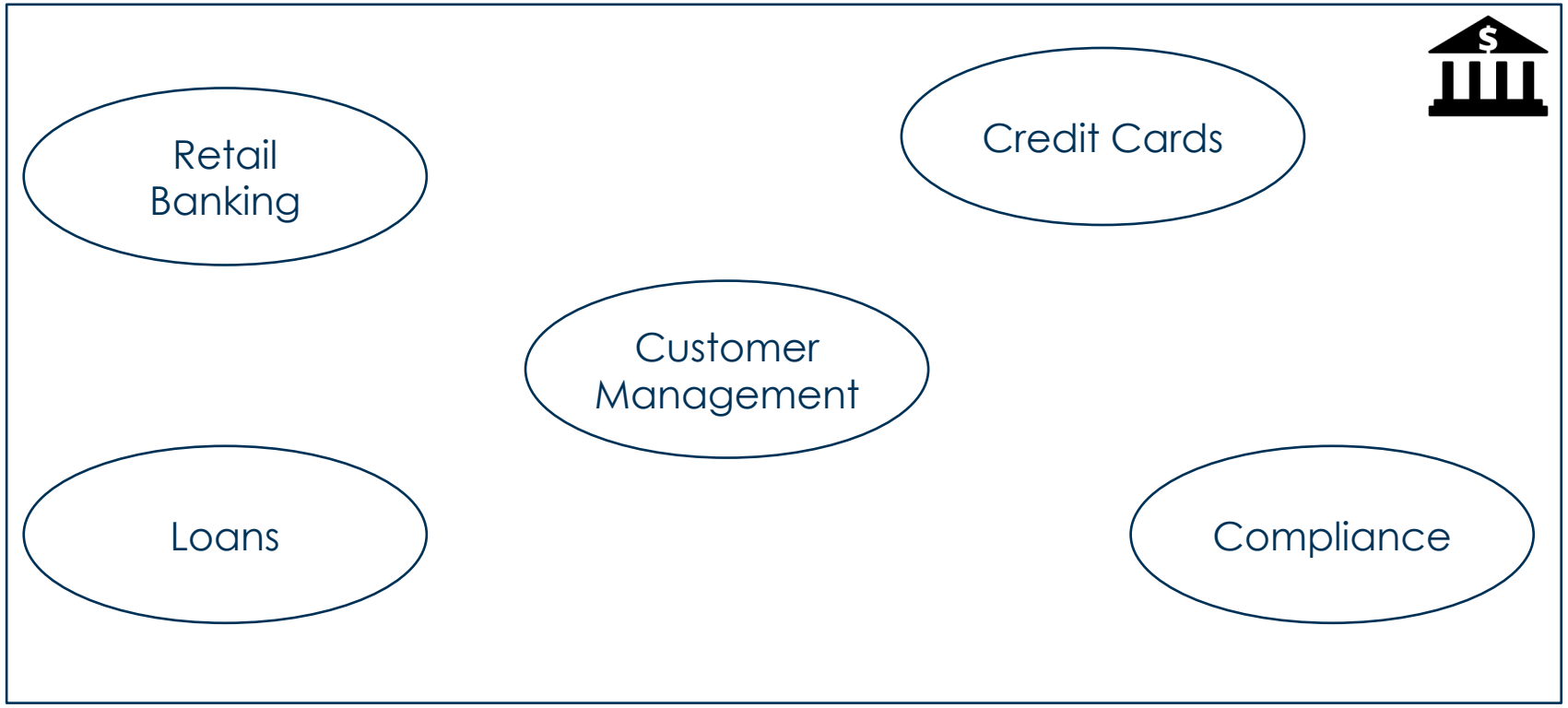
Common model challenges

Characteristics of Bounded Context

Relationship between BC and Domain Model

Simplified Business Domain Model

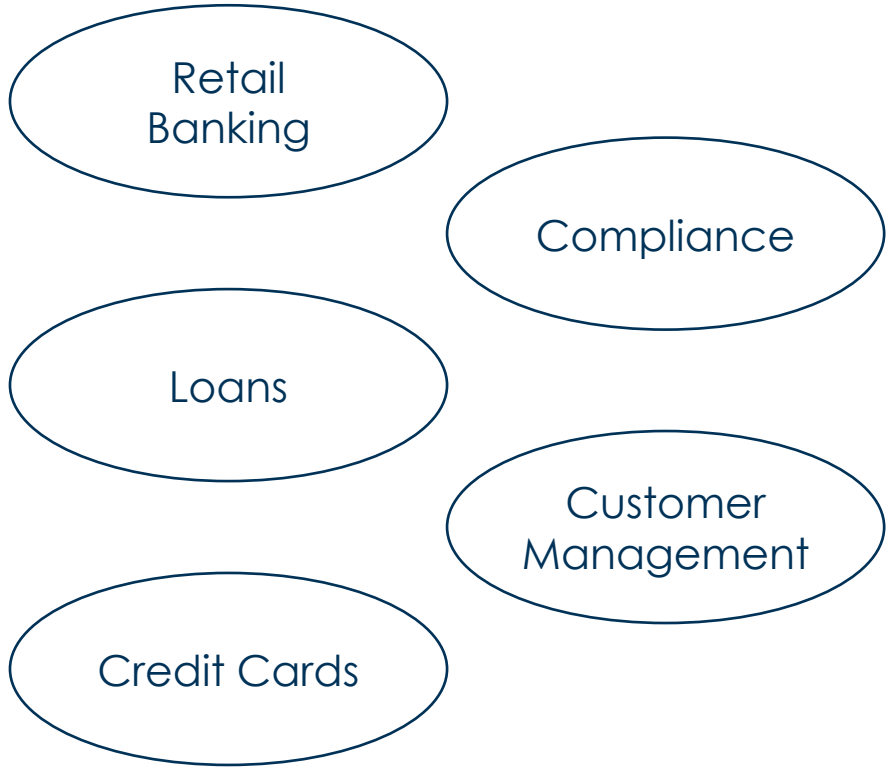
Created based on domain capabilities



Common Domain Model

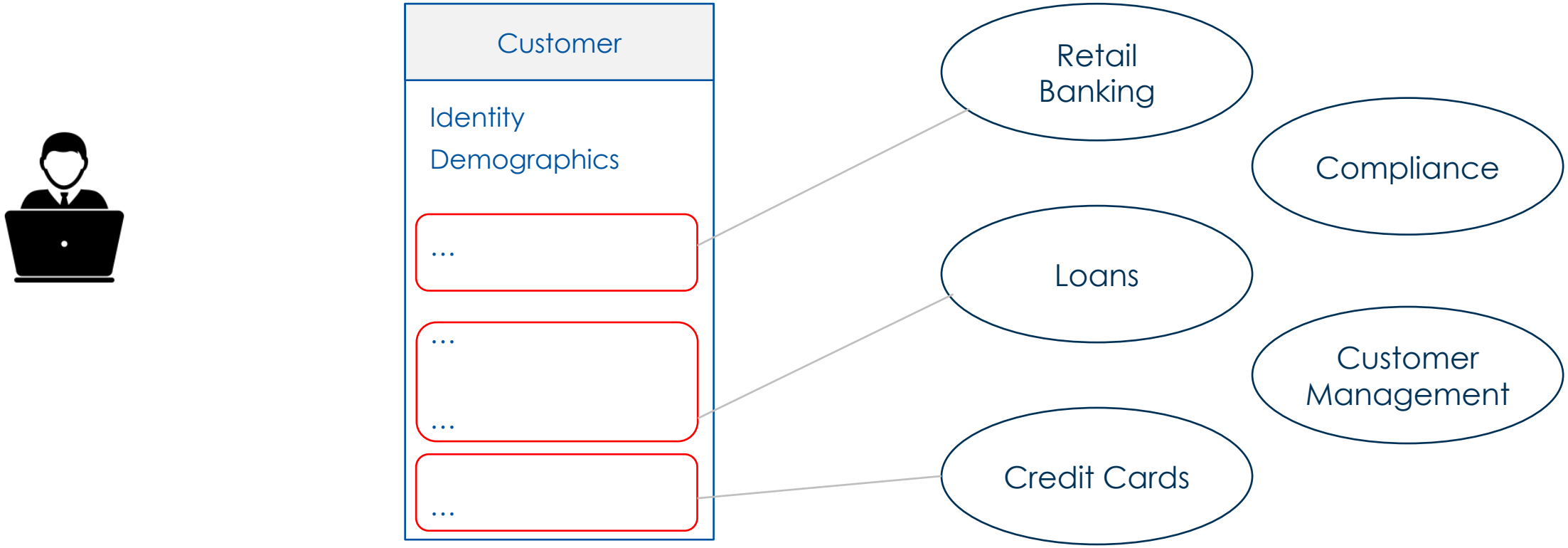
Intended to be used across the enterprise

Customer
Identity Demographics … … … …



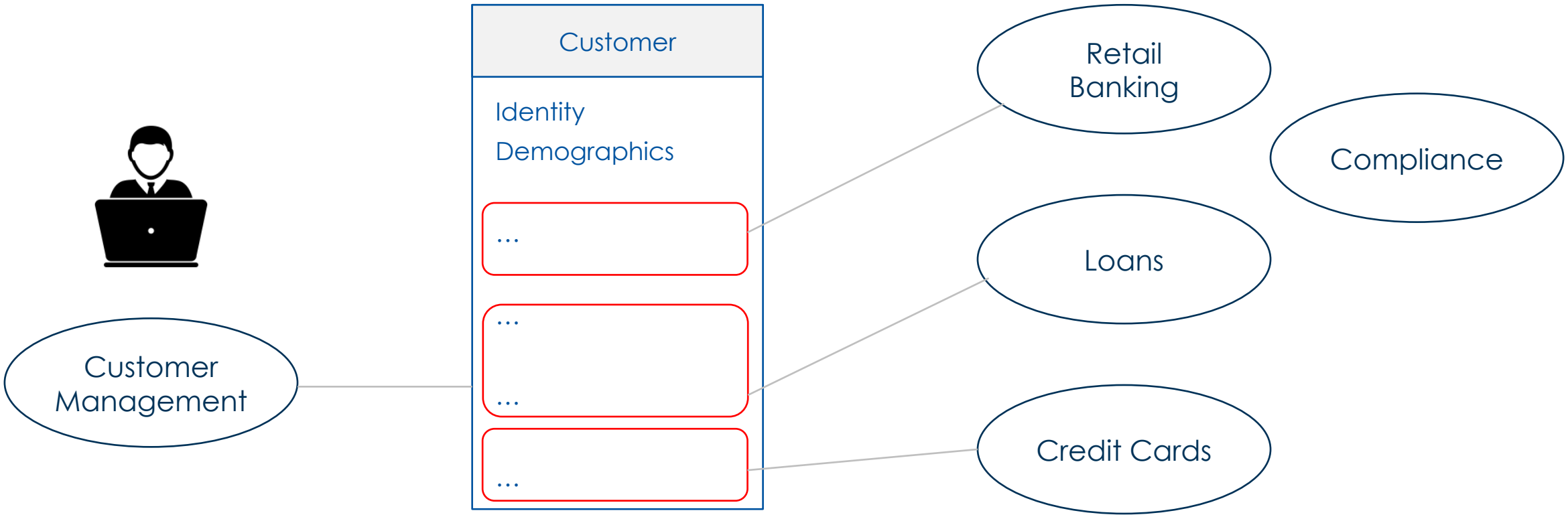
Common Domain Model

Tech Expert gathered the attributes from MULTIPLE experts



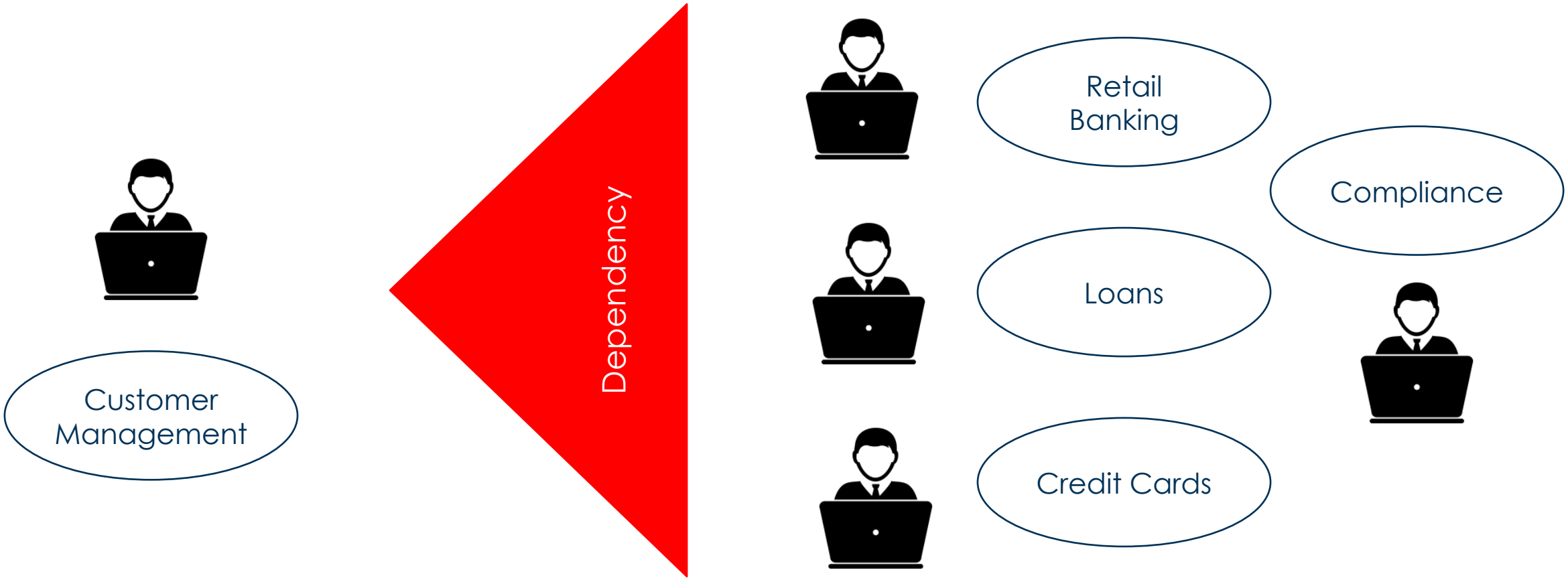
Common Domain Model

One Team takes ownership



Common Domain Model

Creates dependency = Loss of Agility, Conflicts, Complexity …

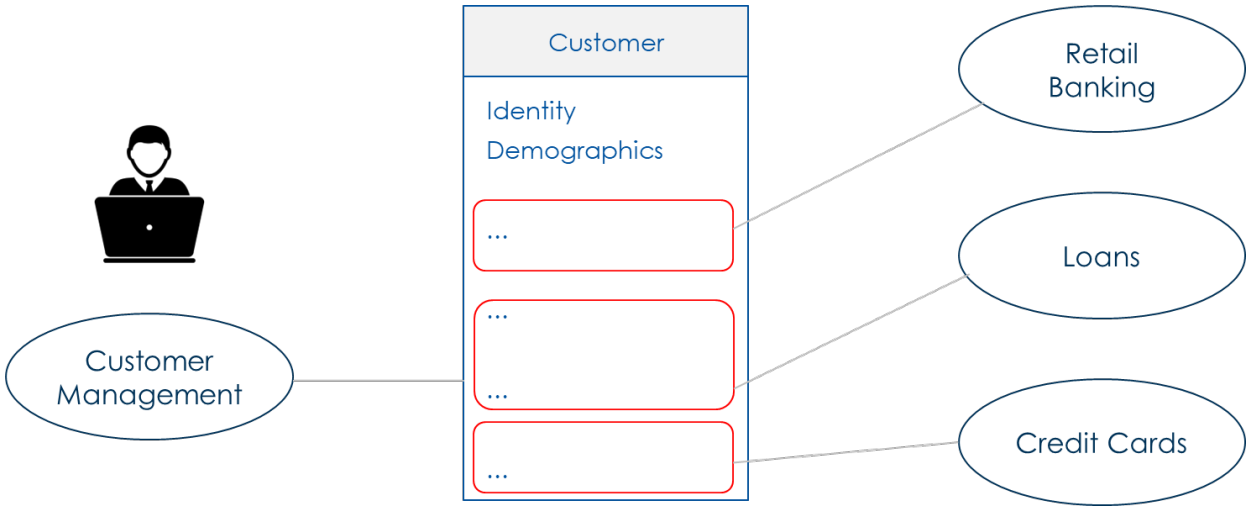


Challenges with Common Domain Model

Difficult to maintain as it cannot satisfy EVERYONE's needs

Models evolve i.e., change ▶ Impacts Everyone

Over a period model loses its value



Make the Business Contexts Independent

> DDD addresses the problems with common domain model by breaking the domain into independent parts referred to as the Bounded Contexts

Complexity

Inherent complexity due to scope and size

Bounded?

Limited to

Within some kind of boundary

Independent?

Free from outside control

Not dependent on another entity

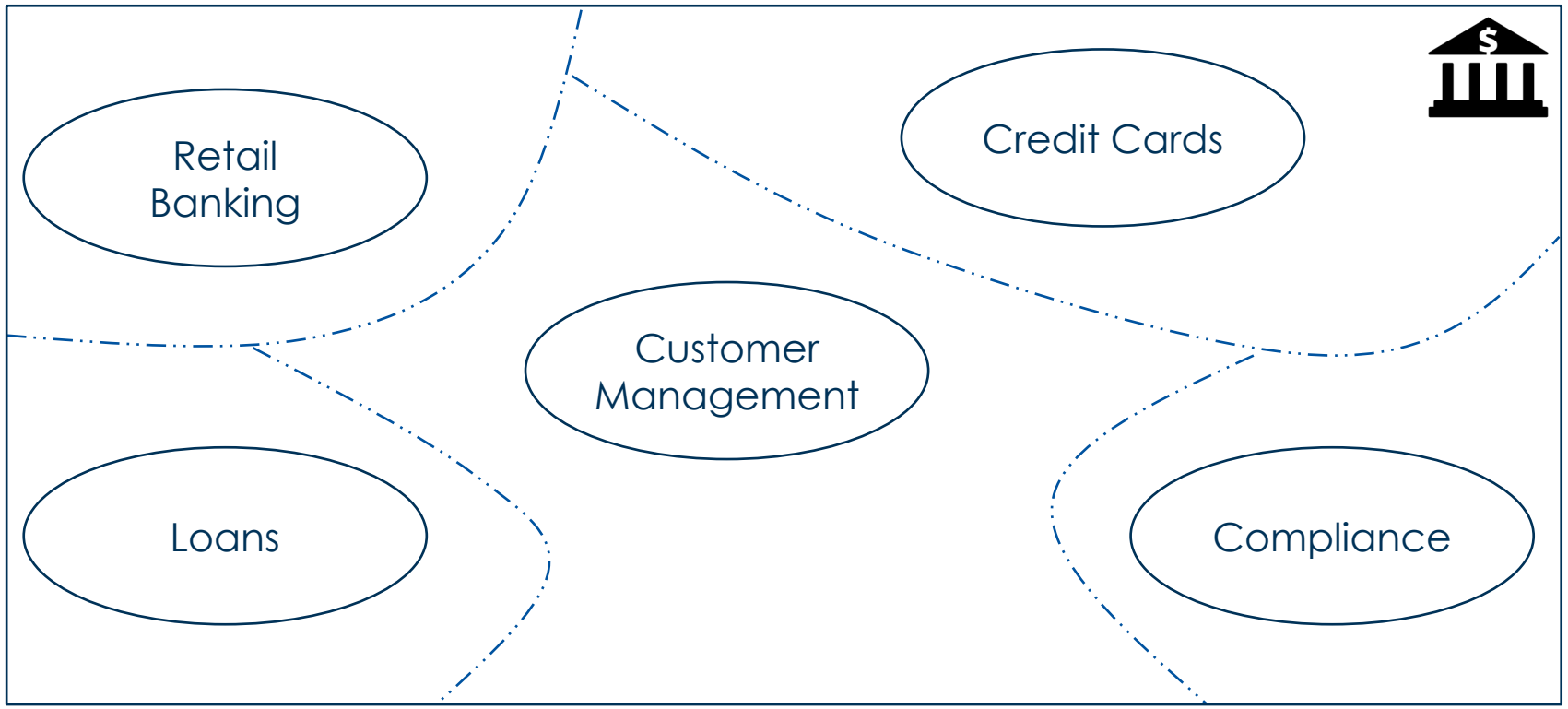
Europe

All these countries are independent with well defined boundaries

- Political & other changes in one country does not have direct impact on other countries
- Laws defined in a country are applicable within the boundary
- Culture, Social norms and Language are well understood within the boundary

Simplified Business Domain Model

Created based on domain capabilities



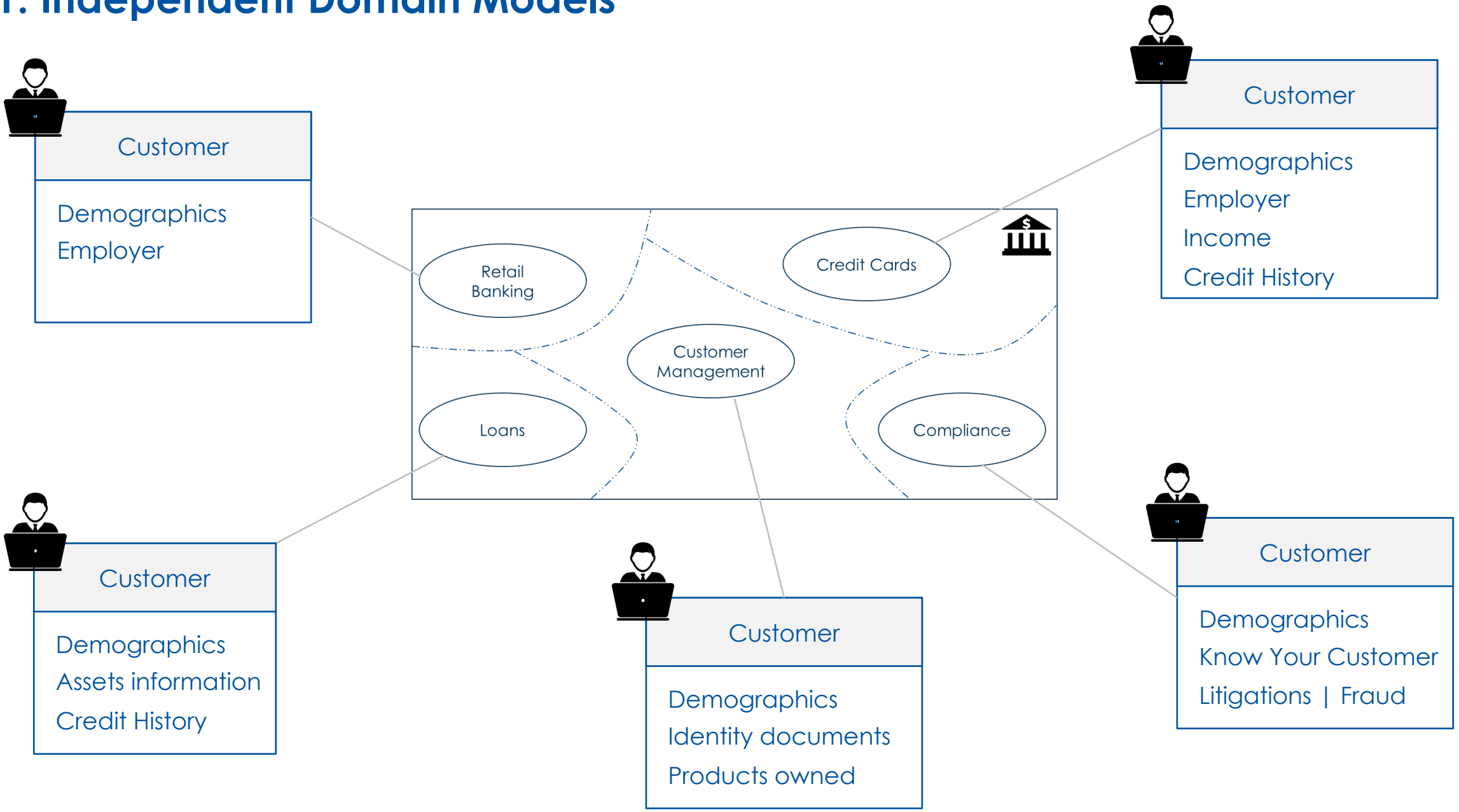
Characteristics of Bounded Contexts

1. Each BC represented with its OWN domain model

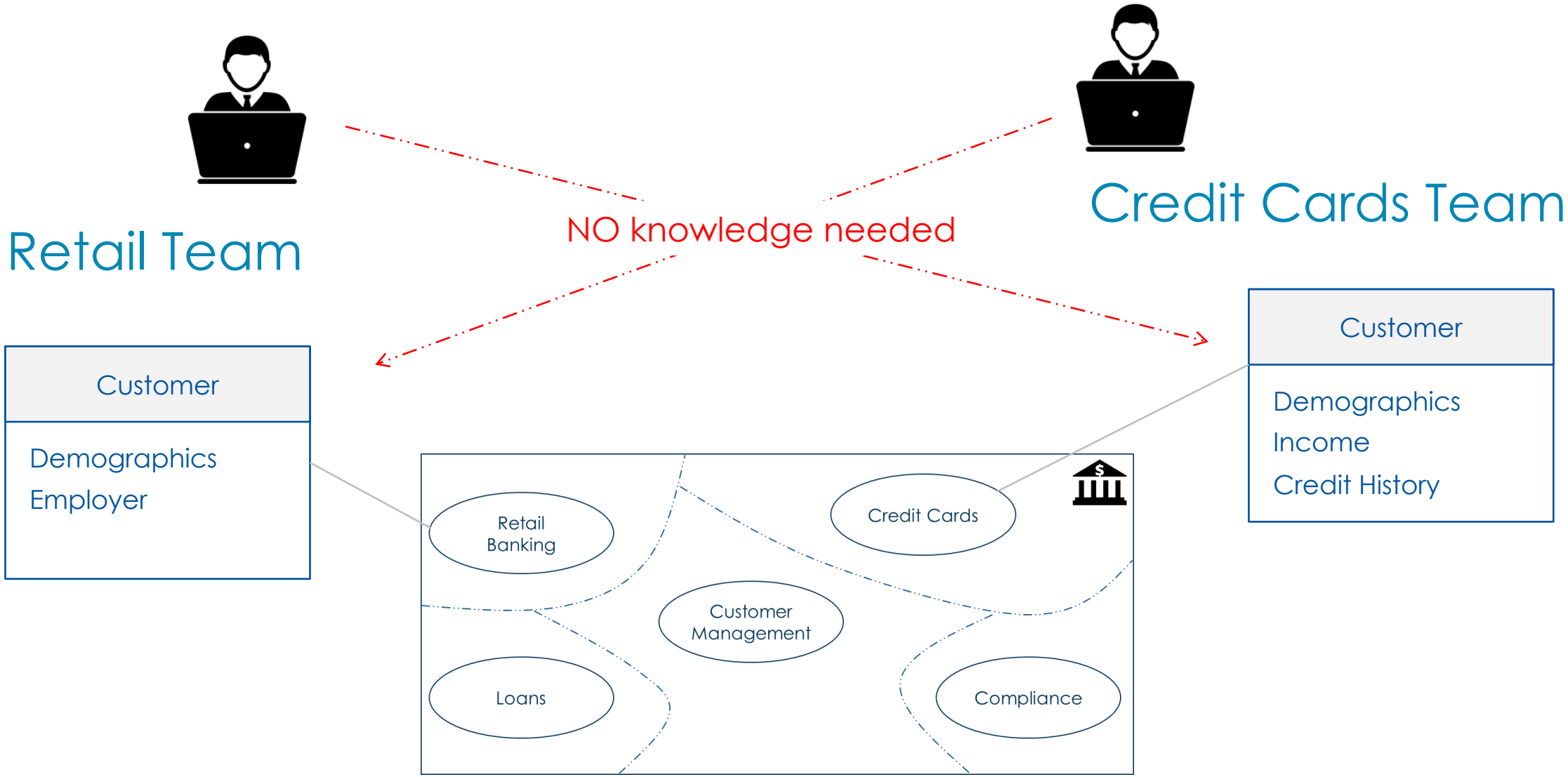
2. Domain model built for a BC applicable ONLY within its boundaries

3. Language used in BC does NOT suffer from Linguistic Challenges

1. Independent Domain Models



2. Model applicable within the BC



2. Model applicable within the BC

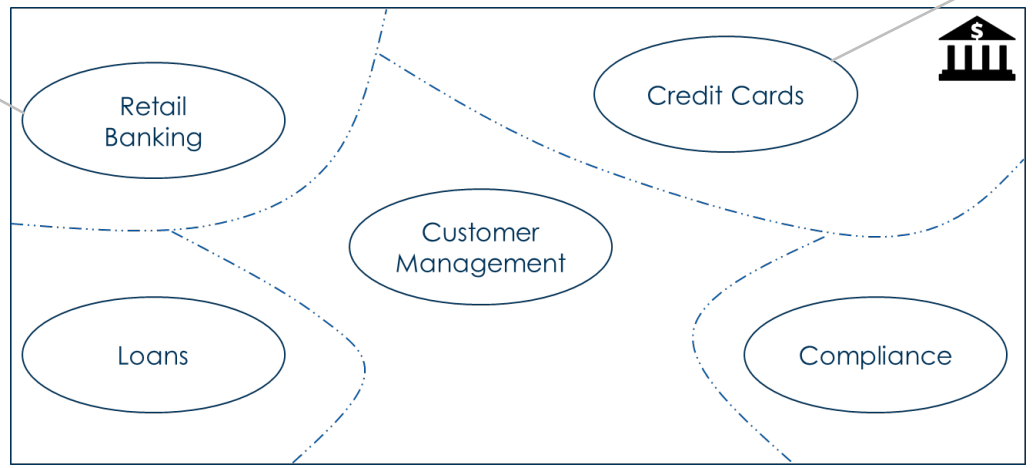
Retail Team

Models can evolve independently

Credit Cards Team

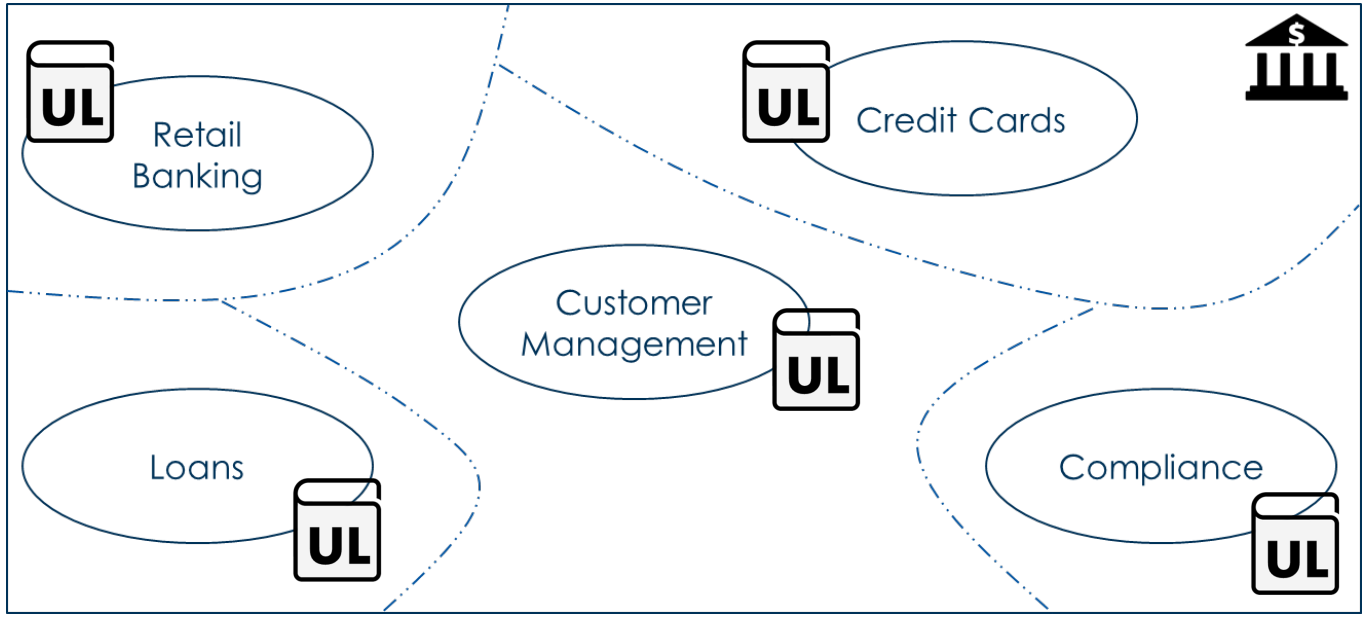
Customer
Demographics Employer

Customer
Demographics Income Credit History



3. Ubiquitous Language & Bounded Context

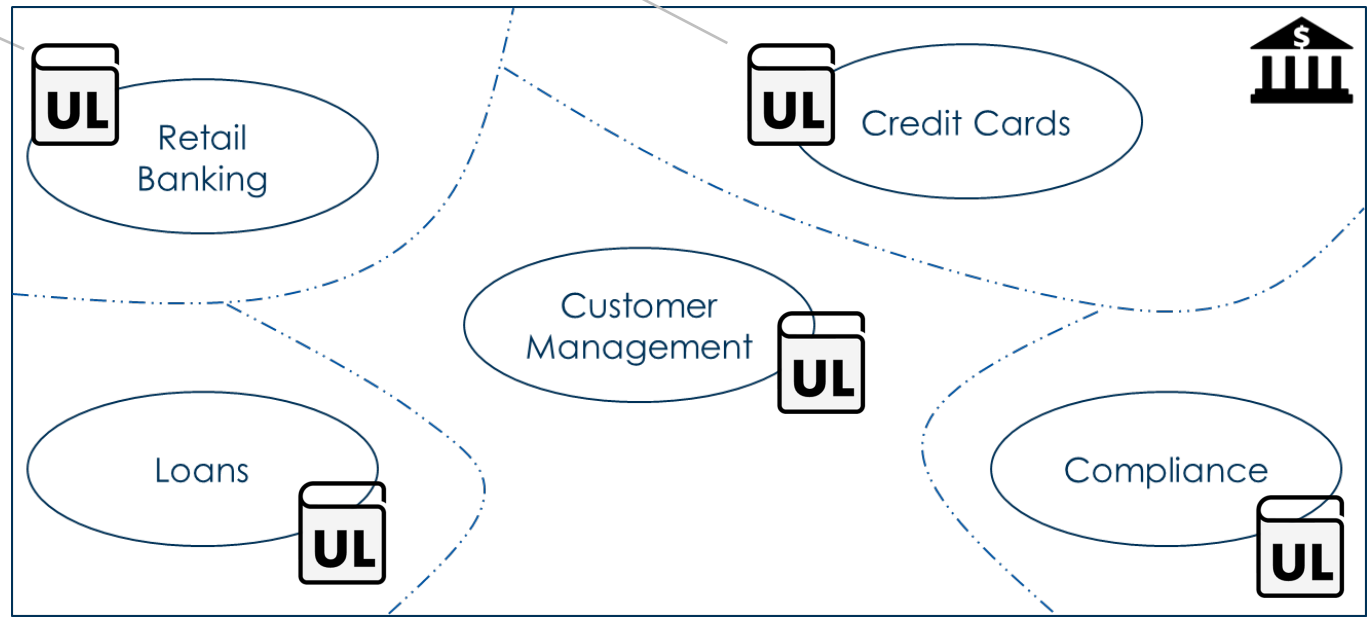
> Each Bounded Context has its own Ubiquitous Language; that is meaningful only within that bounded context



3. Ubiquitous Language & Bounded Context

Retail Team

Credit Cards Team



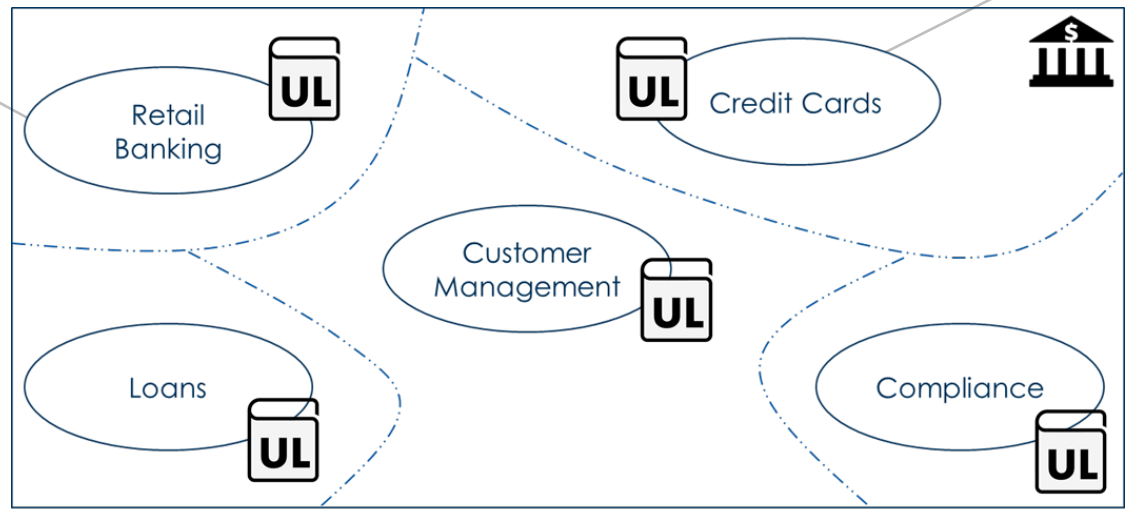
Each Team uses its own UL

Retail Team

Credit Cards Team

Customer
Demographics Employer

Customer
Demographics Income Credit History



Quick Review

- DDD suggests breaking the problem space into Bounded Contexts

1. Models developed independently within each BC

2. Model applicable within the BC

3. Each BC has a Ubiquitous Language

Discovering the Bounded Contexts

An art rather than science

1. Guidance on discovery of BC
2. Clues to identifying the BCs

Bounded Context Discovery

There is **NO** silver bullet!!!

“ *It's an Art not a Science*

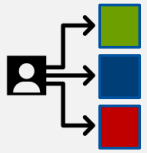
How to setup the Bounded Contexts?

Guidance based on *Experience*

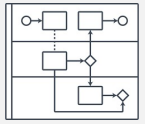
Art of discovering the Bounded Contexts

Leverage the existing assets; Partner with the Domain experts

Organization structure

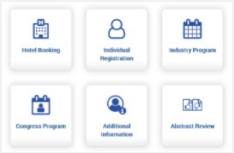


Business Expert's Responsibilities



Key Activities

Business Language i.e., the linguistic clues

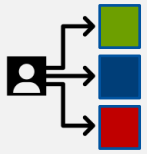


Existing monolithic applications | Modules

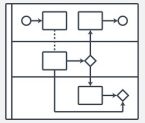
Art of discovering the Bounded Contexts

Leverage the available assets & collaborate with the experts

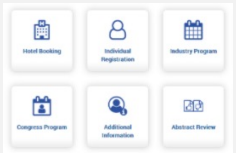
Organization structure



Business Expert's Responsibilities



Key Activities



Existing monolithic applications | Modules

Pay attention to the clues to demarcate the boundaries

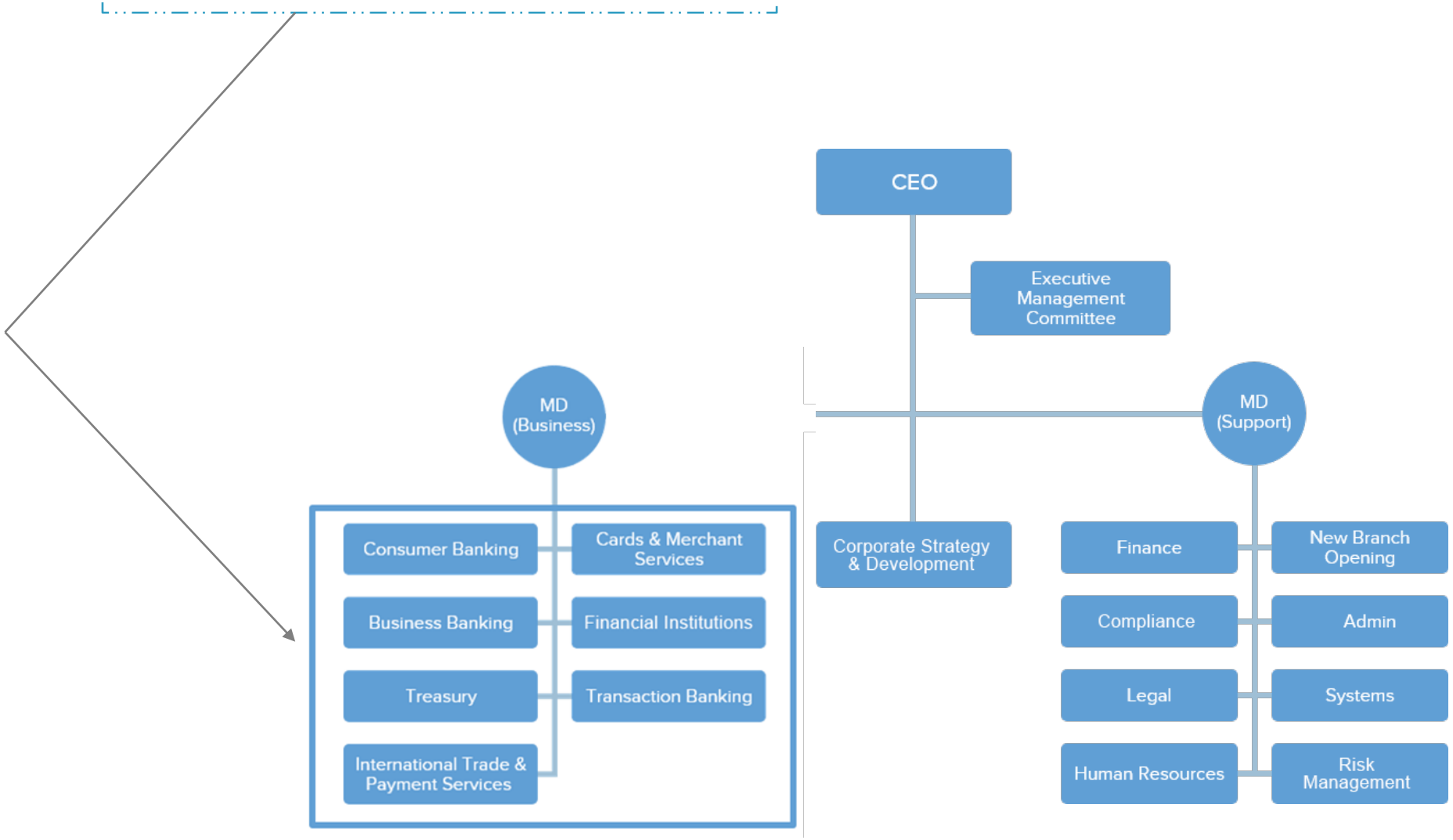
Do NOT get hung up on finding the perfect BC

Do NOT get hung up on finding the perfect boundaries for the bounded contexts in your domain

OK to start with some boundary than None ☺

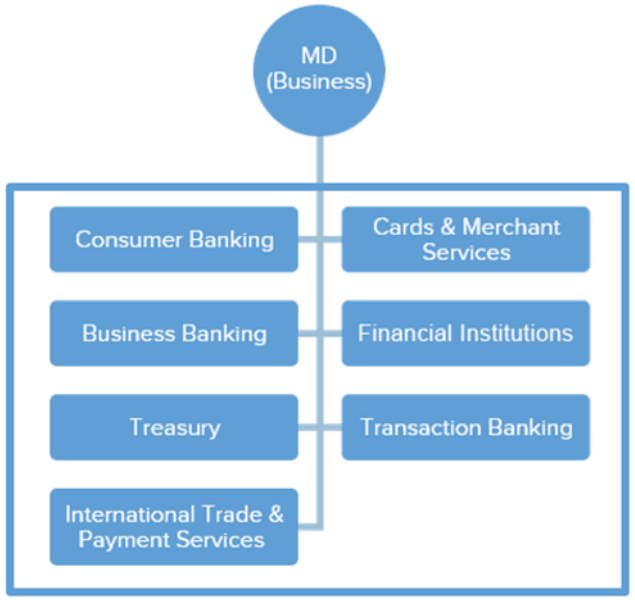
Organization Structure

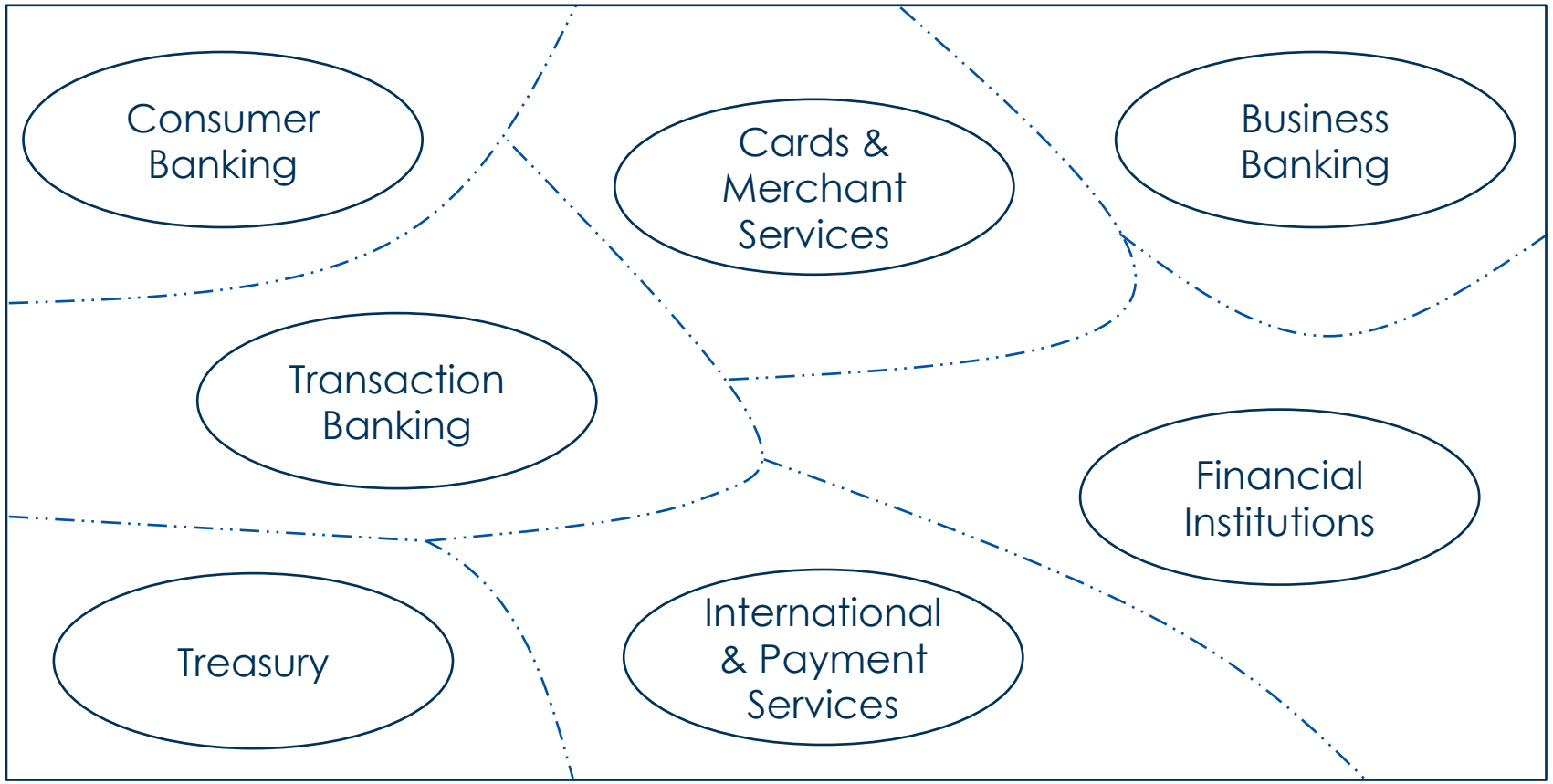
Pay attention to Business Function areas



Business Capabilities

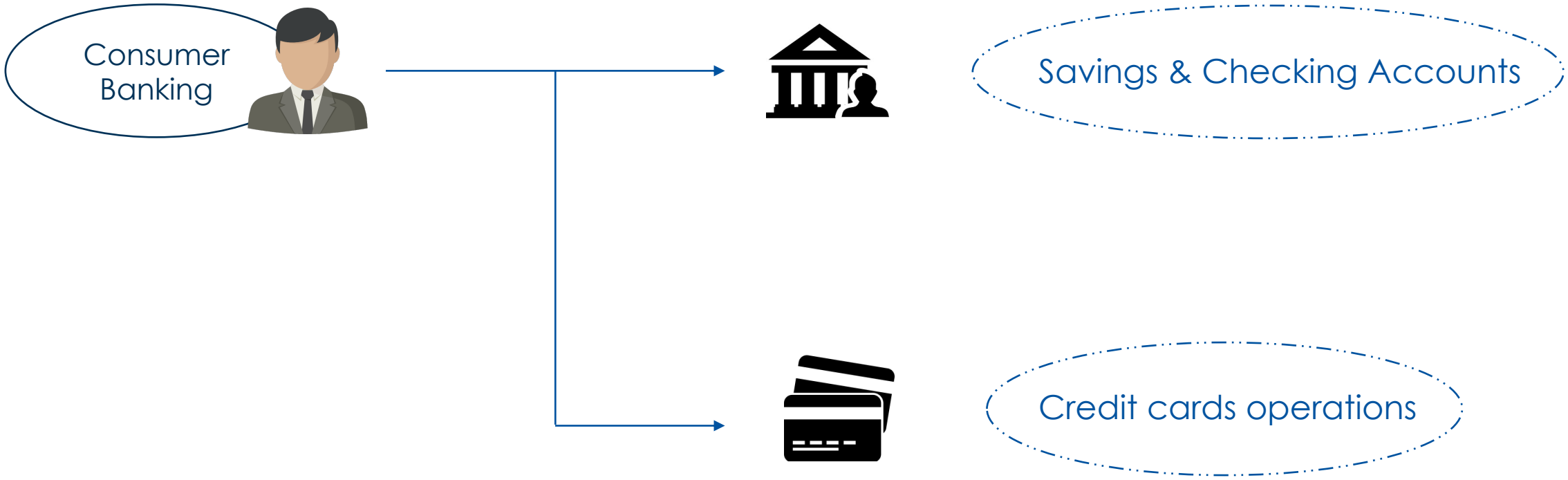
Start by drawing the boundaries around business functions





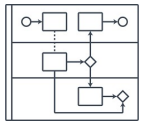
SME Responsibilities

Experts may be responsible for multiple functions



Build the business glossary & look for linguistic clues

Key Activities



Makes the business model work

Consumer Banking

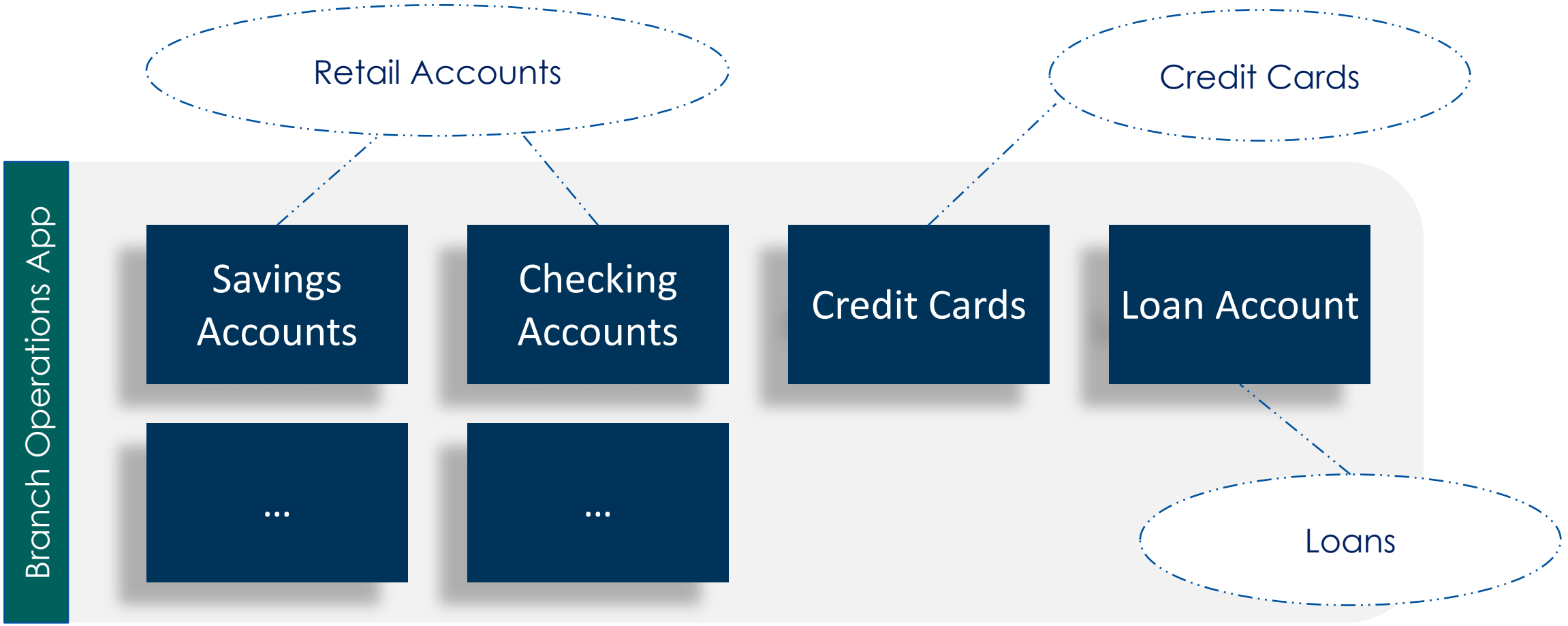
- Customer Support Operations
- Accounts Management
- Branch Operations

IT Expert & Business Expert chalks out the details of the key activities

Build the business glossary & look for linguistic clues

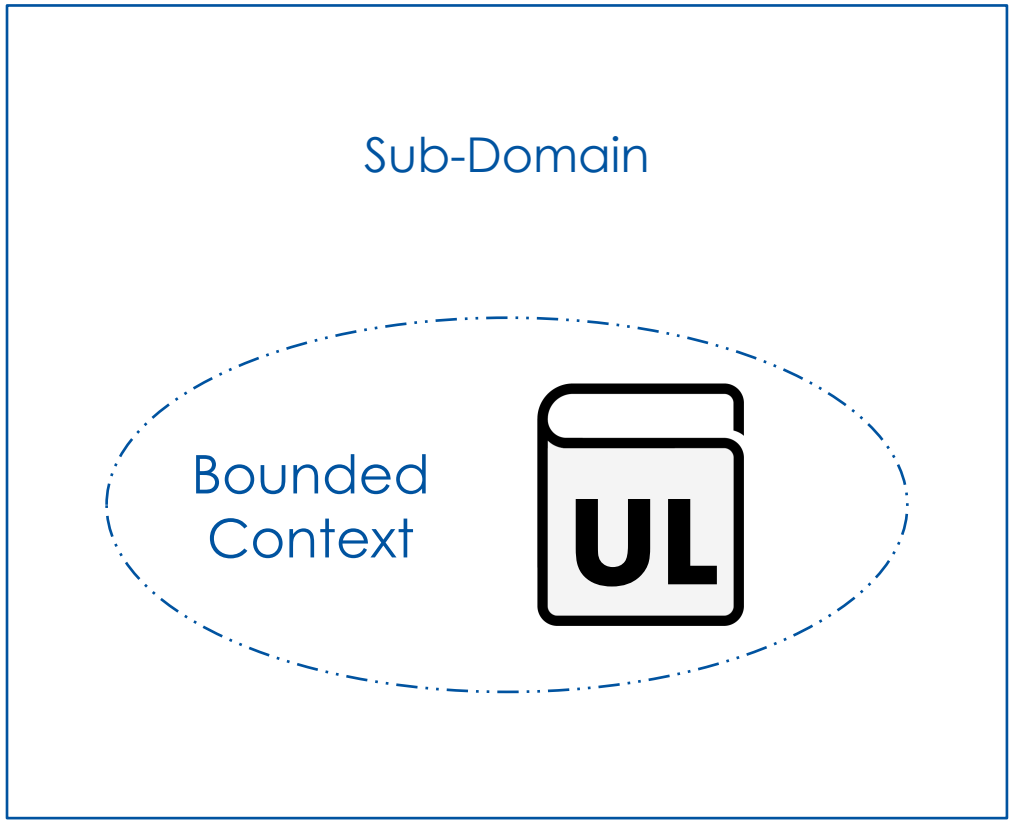
Existing Application | Modules

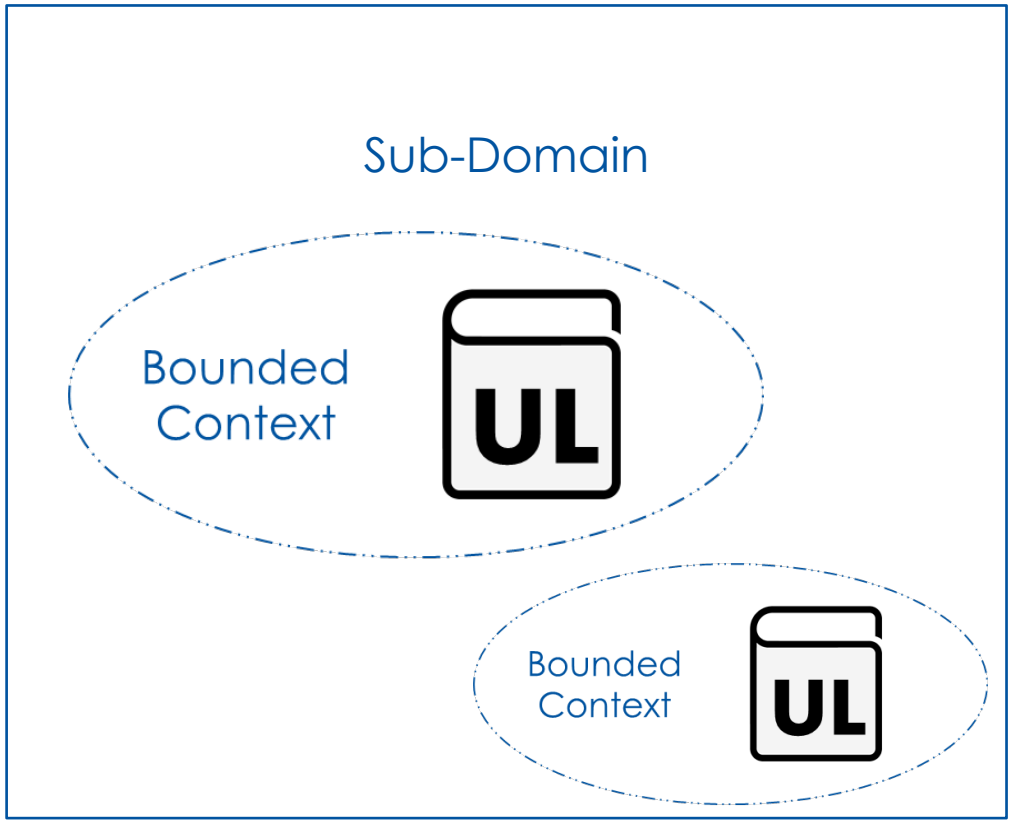
Existing application | modules may hold clues to boundaries



Bounded Context | Sub-Domain

Sub Domains may have 1 or more BC

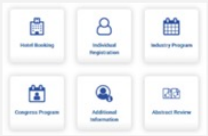




Quick Review

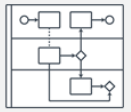
- Existing assets & business expert partnership

Organization structure



Existing monolithic applications | Modules

Business Expert's Responsibilities



Key Activities

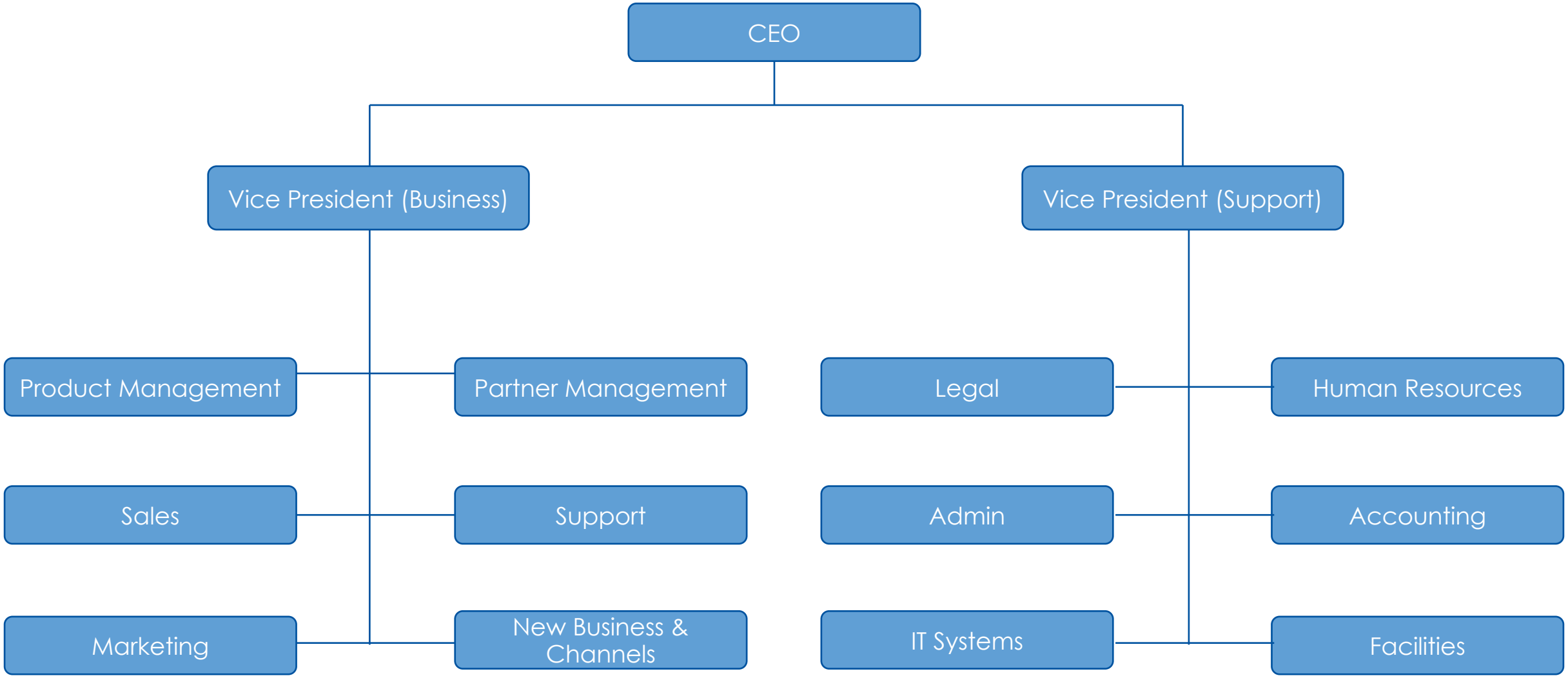
Business Language i.e., the linguistic clues

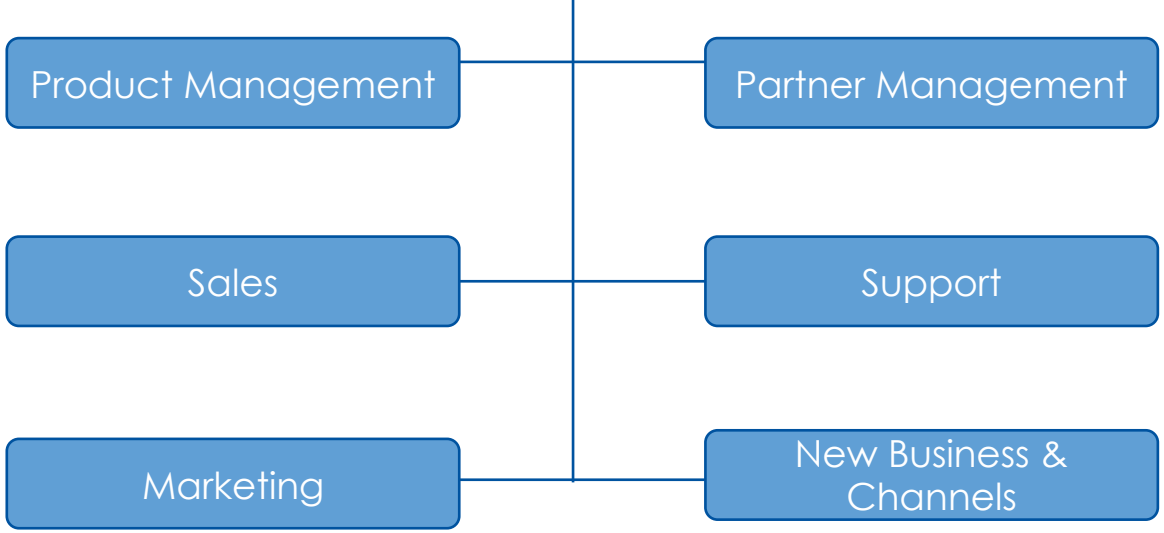
Exercise : Drawing the Boundaries

Discovering the BC for ACME

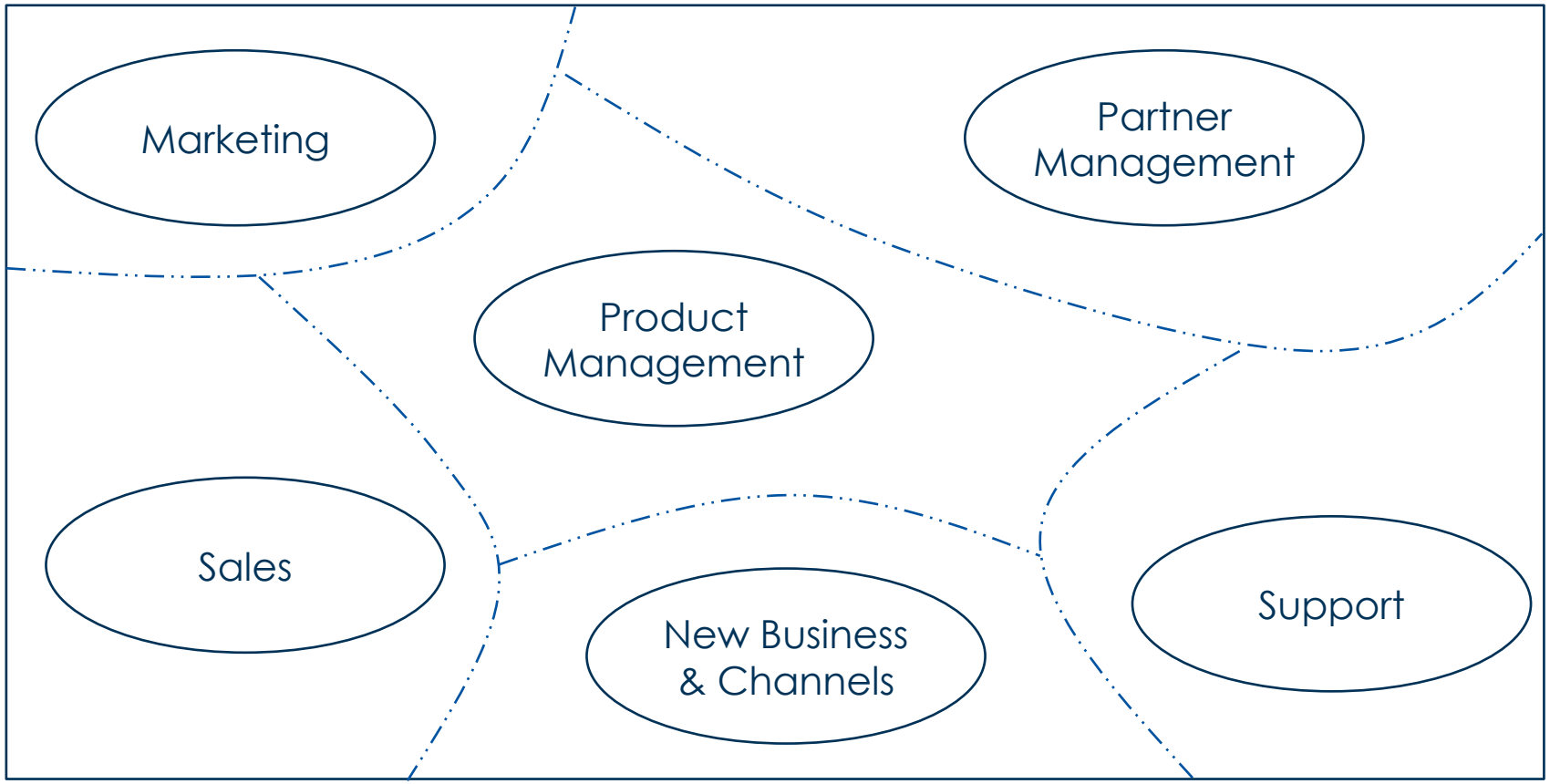
1. Setup Bounded Contexts based on capabilities
2. Go over the *Key Activities* for Support
3. Analyze the language clues from support

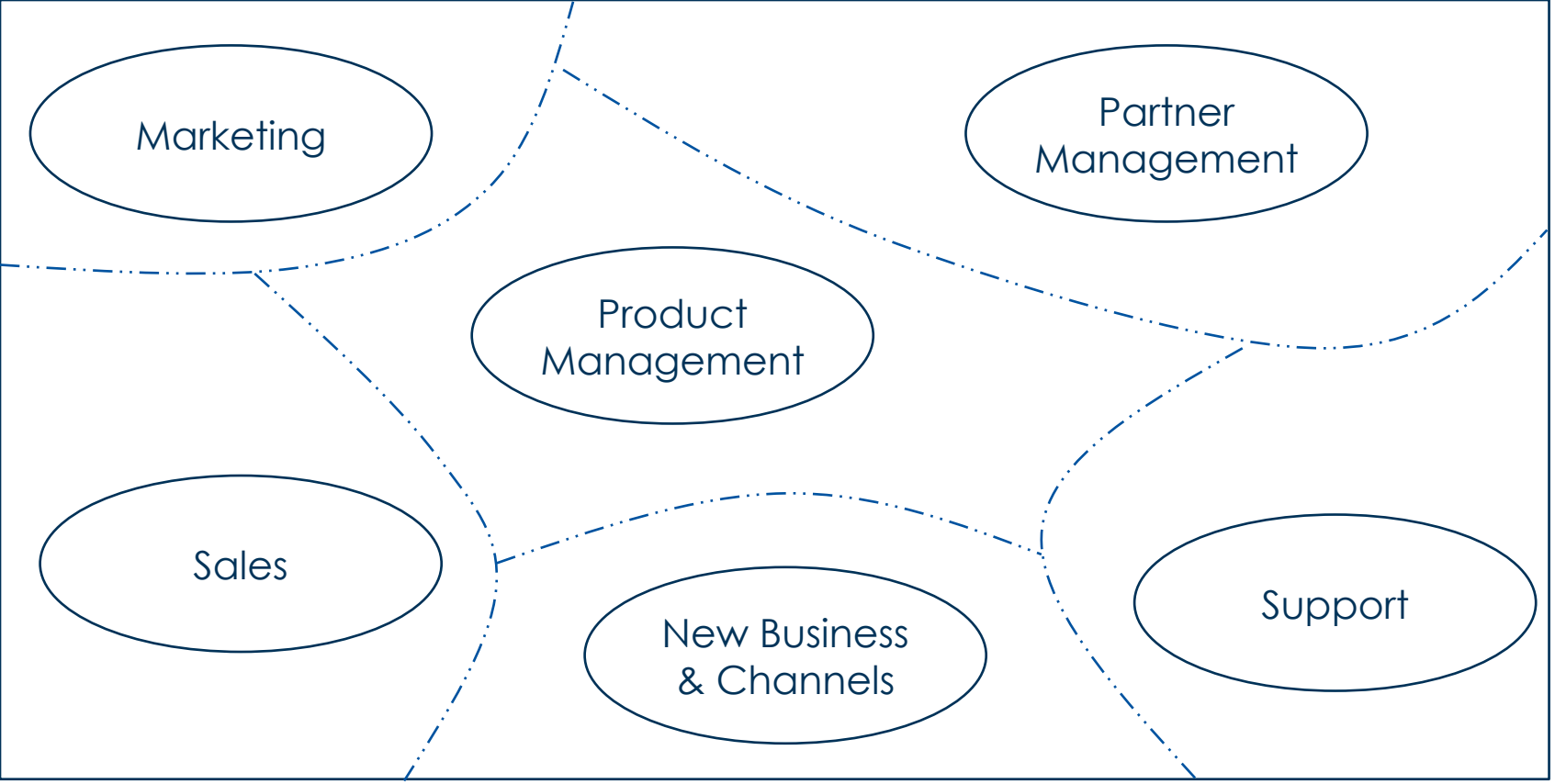
ACME Organization Structure




Product Management
Partner Management
Sales
Support
Marketing
New Business & Channels

ACME Travel


Marketing
Partner Management
Product Management
Sales
New Business & Channels
Support



ACME Travel

Kathy, Support

Responsible for supporting Customers and Partners

Support function at ACME

Responsible for supporting Customers and Partners

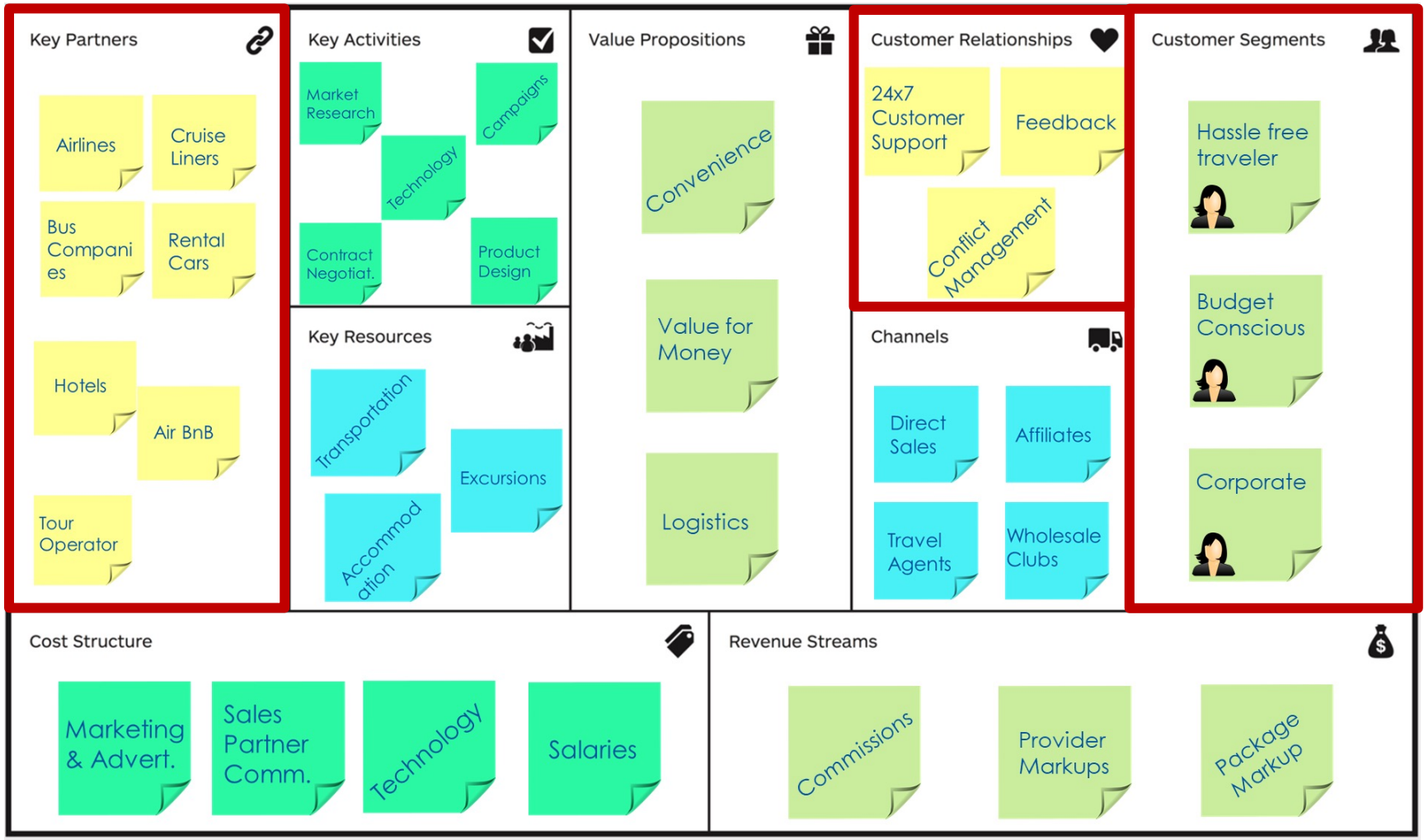
Kathy, Support

Customer Support

24x7

Partner Support

8:00 AM to 5:00 PM ET
Monday - Friday



Support | Key Activities

24 x 7

Customer

Call Center

Kathy, Support

Providers

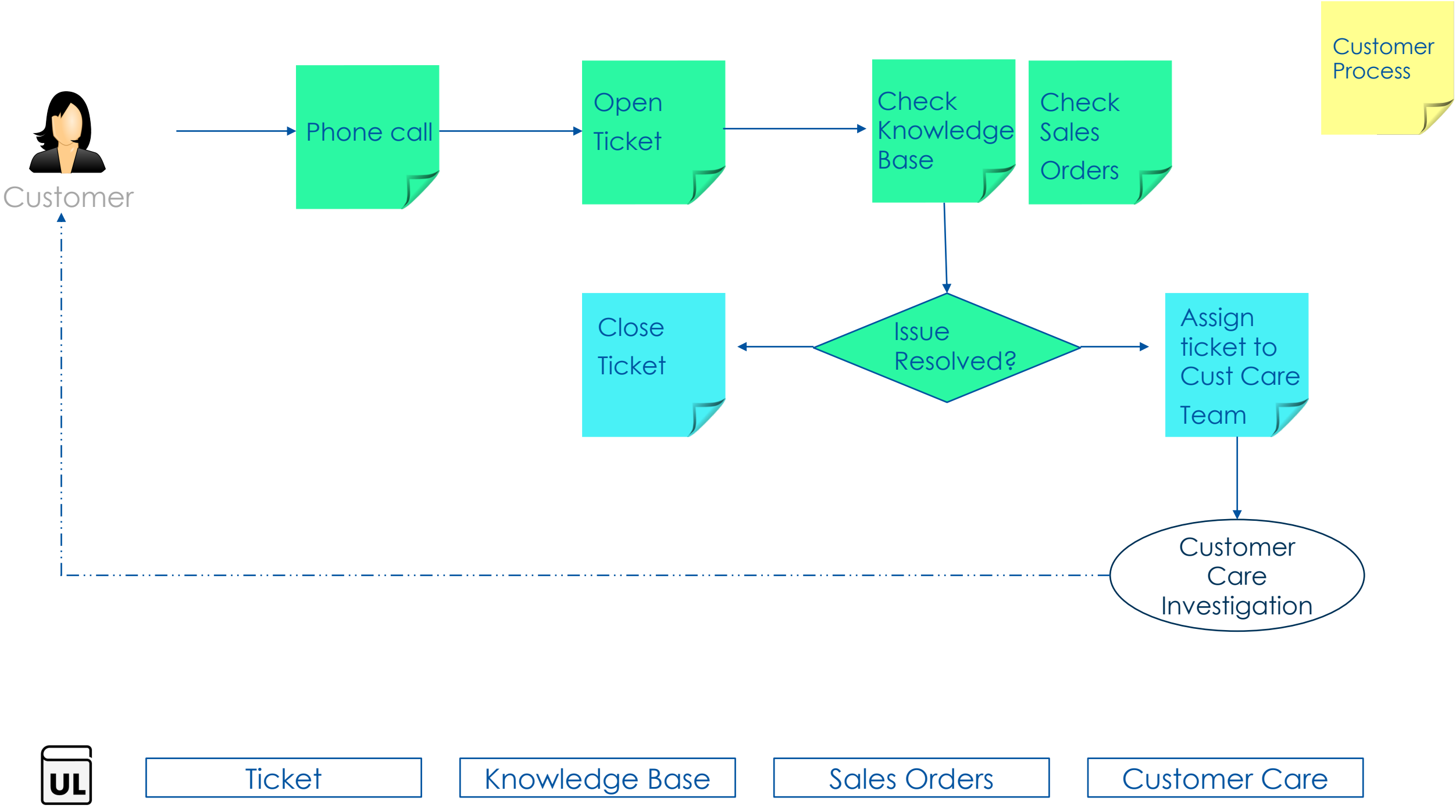
8:00 AM to 5:00 PM ET
Monday - Friday

(off hours support not available)

Customer Process

Provider Process

Key activities


Customer Process
Customer
Phone call
Open Ticket
Check Knowledge Base
Check Sales Orders
Close Ticket
Issue Resolved?
Assign ticket to Cust Care Team
Customer Care Investigation
UL
Ticket
Knowledge Base
Sales Orders
Customer Care

Customer Process

Customer

Phone call

Check Ticket Status

Check the Ticket status

Ticket

Knowledge Base

Sales Orders

Customer Care

Provider Process

Providers

Phone call

Open Provider Case

Check Knowledge Base

Check Provider Contract

Issue Resolved?

Close Provider Case

Assign ticket to App. Team

Partner Management

Accounting

UL

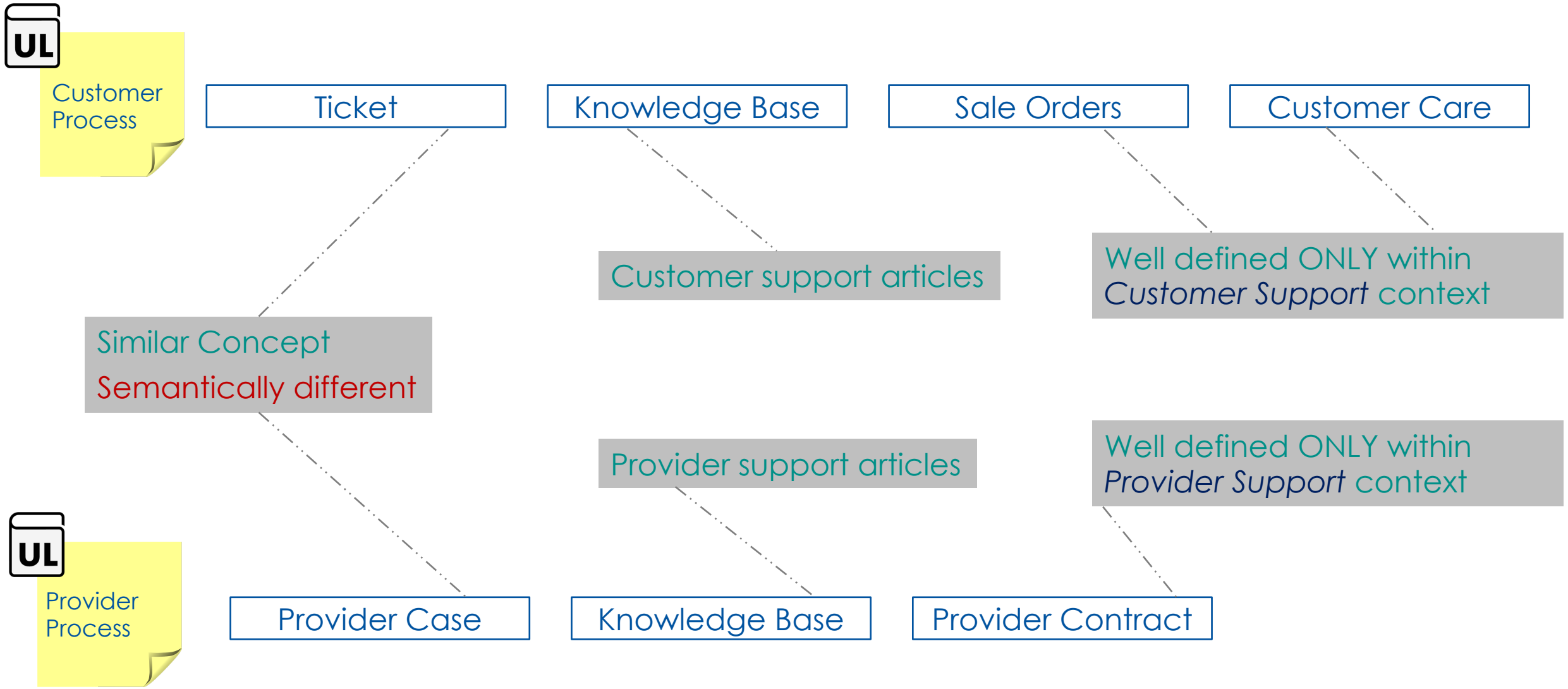
Provider Case a.k.a. Partner Case

Knowledge Base

Provider Contract

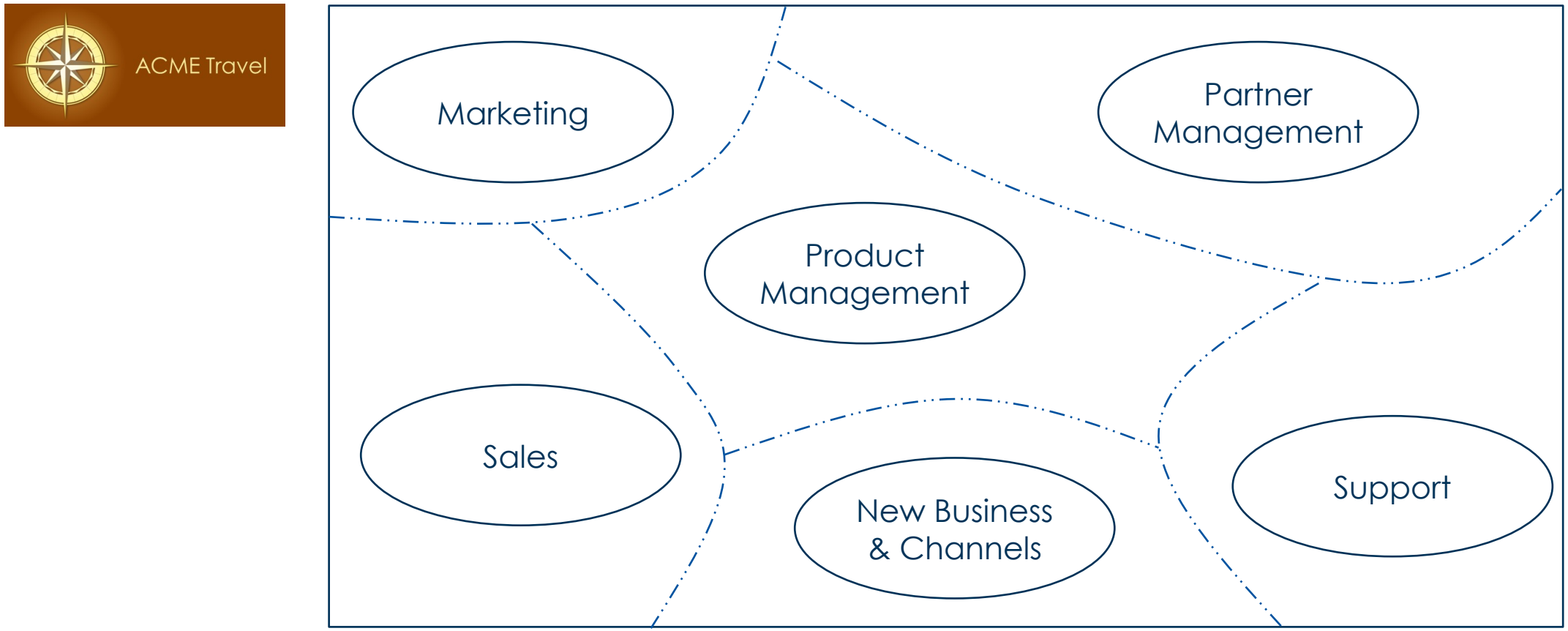
Draw a rough sketch of the Business Capabilities in your organization

Support | Business Language



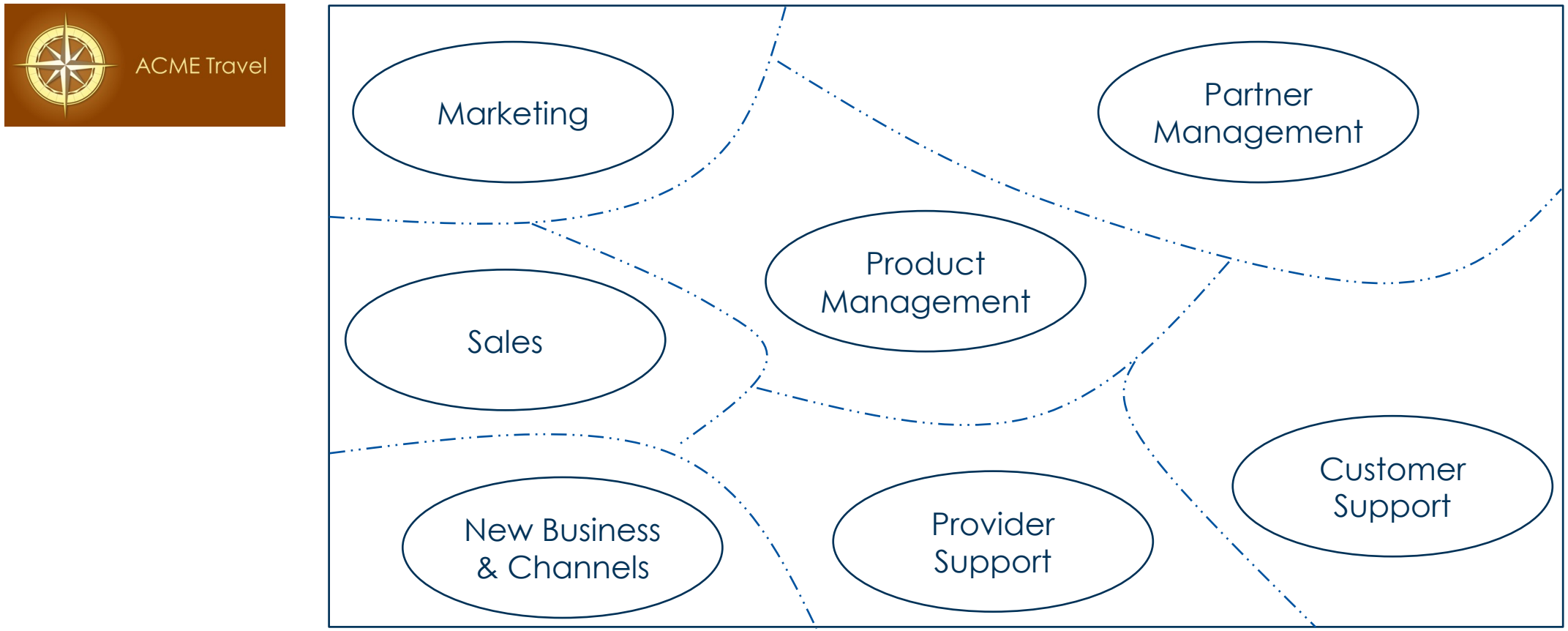
Business Capabilities | Organization structure

Drawing the boundaries based on domain capabilities



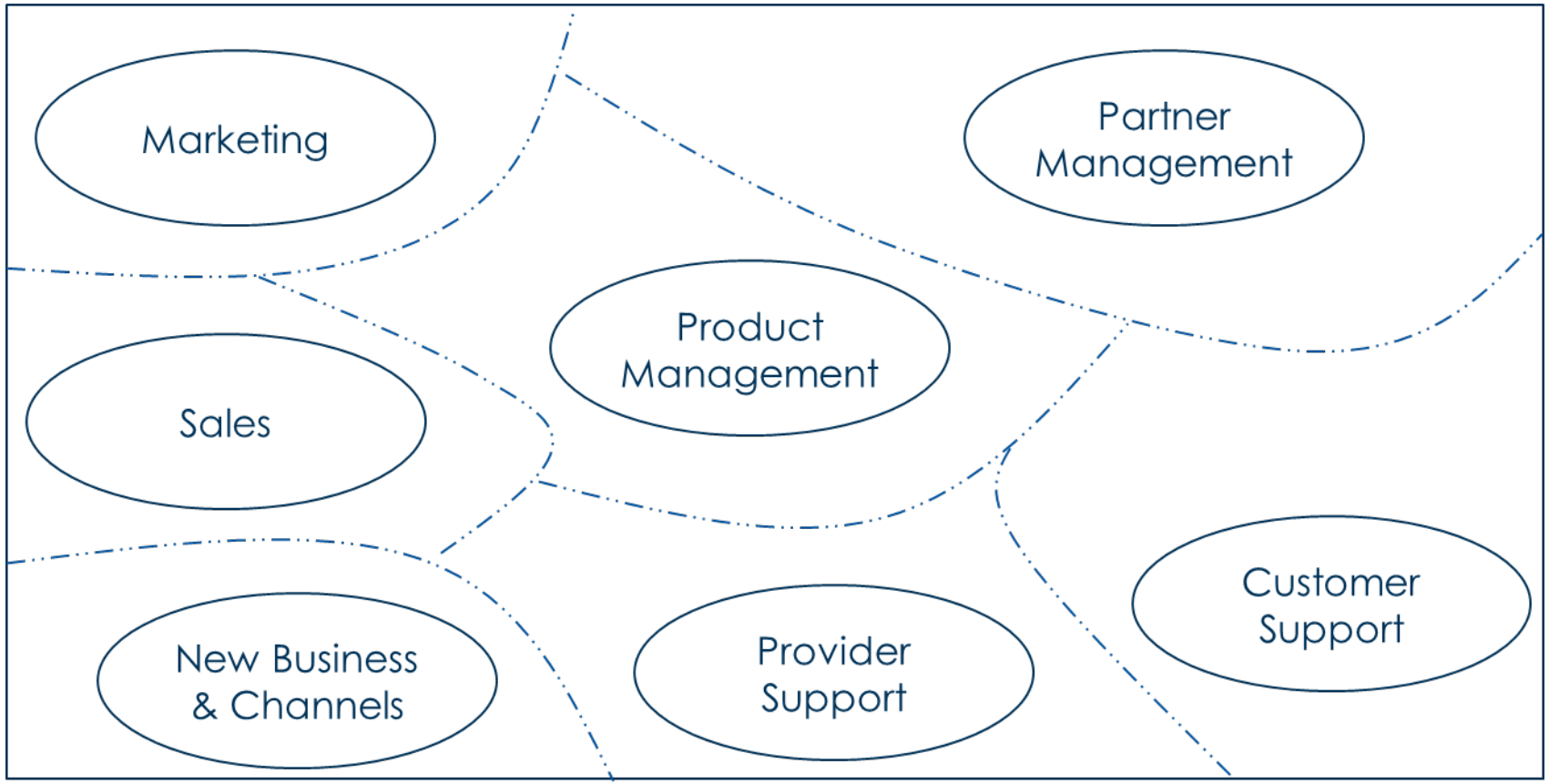
Business Capabilities | Organization structure

Support context split into 2 independent Bounded Contexts

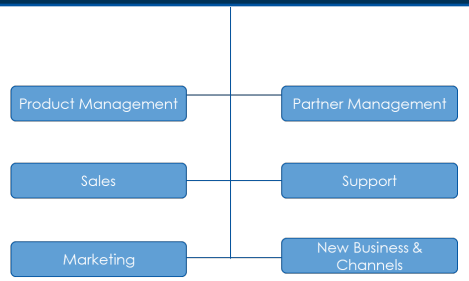


Business Capabilities | Organization structure

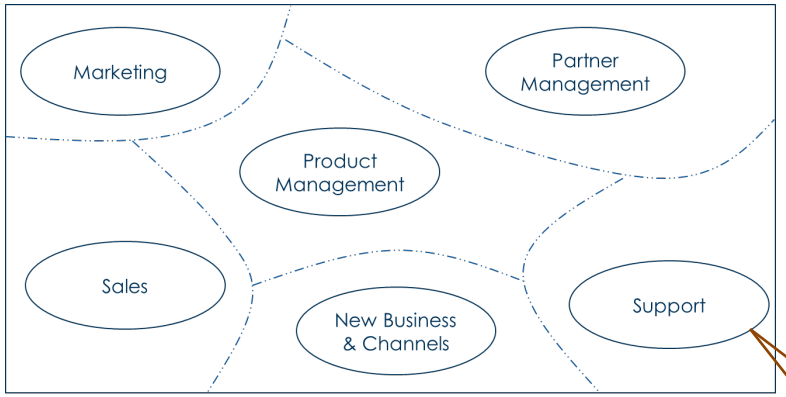
Support context split into 2 independent Bounded Contexts



Quick Review



ACME Org Structure



ACME Travel

Analyzed Support Activities

Analyzed Business Terms