



Modelado y Validación de Arquitectura

German Alonso Suárez Guerrero

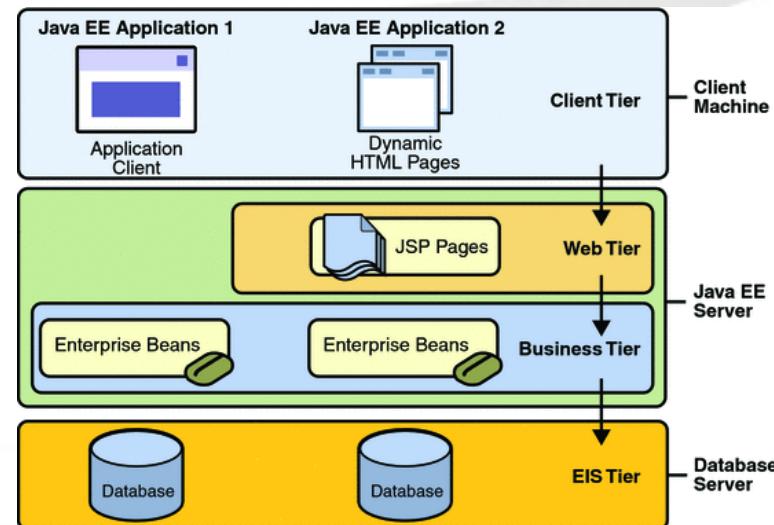
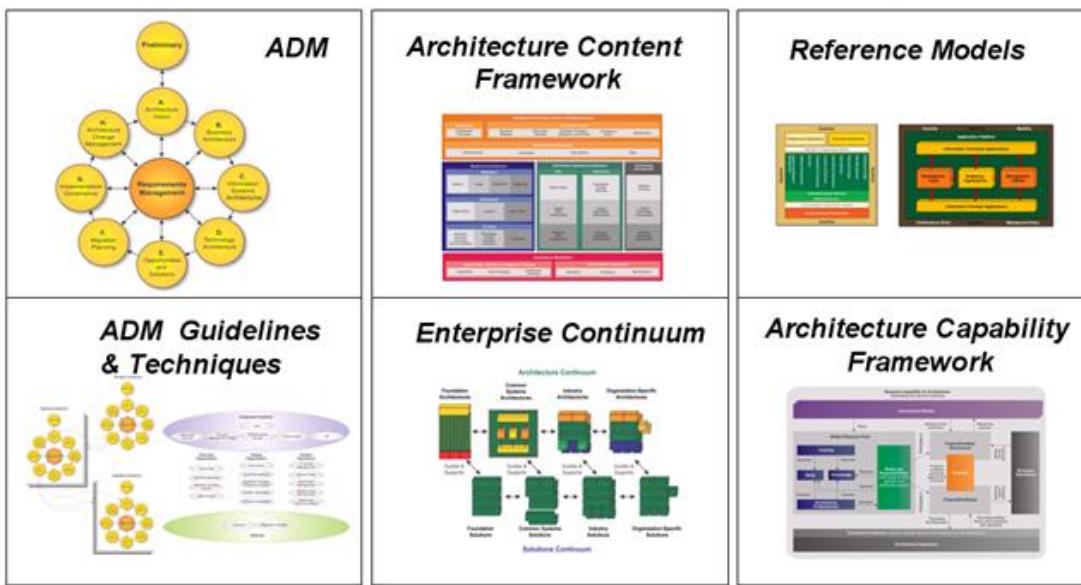
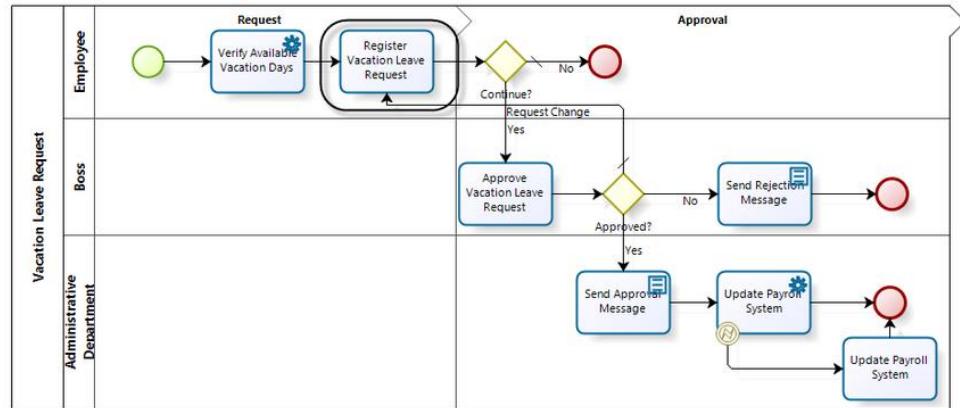


Reflexiones

- ¿Qué temas deberían estar en esta materia?
- ¿En qué temas del punto anterior puedo aportar con mi experiencia laboral?
- ¿Qué actividades hace un arquitecto empresarial de software?
- ¿Qué habilidades debe tener un arquitecto empresarial de software?
- ¿Cuál es el límite entre Diseño de Software y Arquitectura de Software?
- ¿Cuándo está bien hecha una Arquitectura Empresarial de Software?



Convergencia





Estrategia de Negocio





Estrategia de Negocio

- Strategy is the pattern of decisions in a company that determines and reveals its objectives, purposes, or goals, produces the principal policies and plans for achieving those goals, and defines the range of business the company is to pursue, the kind of economic and human organization it is or intends to be, and the nature of the economic and non-economic contribution it intends to make to its shareholders, employees, customers, and communities.

Kennet, Andrews



Estrategia de Negocio

- Misión estratégica – Propósito ¿Qué quiere alcanzar la organización? Alcance de la operación (Productos, Mercados, Geografía)
- Plan estratégico – Modelo o Plan de Negocio. ¿Cómo la organización cumple su misión?
- Acciones estratégicas – Acciones que toma la compañía para ejecutar su plan estratégico y alcanzar su misión estratégica.



Rivalidad Corporativa



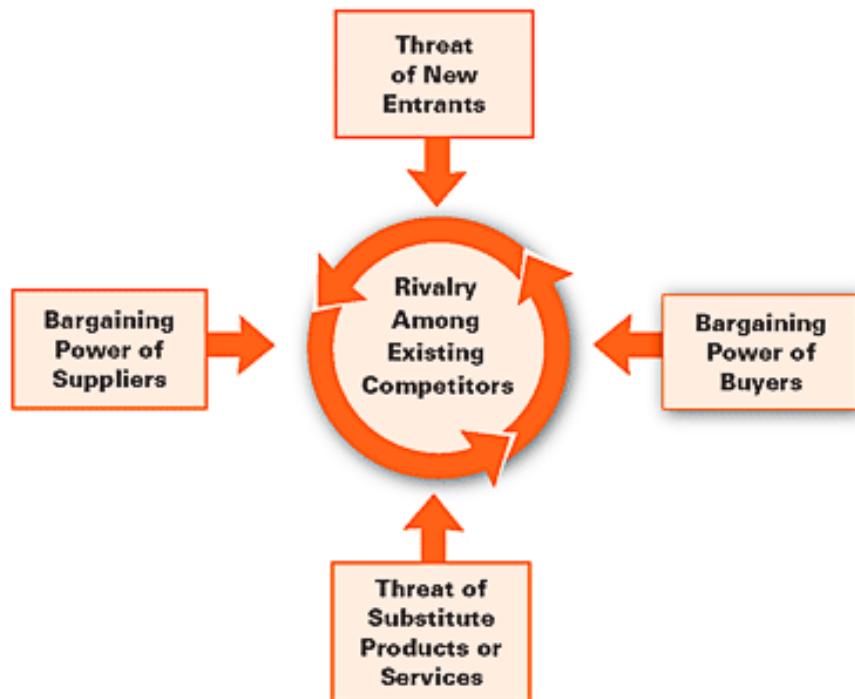


Análisis Estratégico

ANÁLISIS FODA

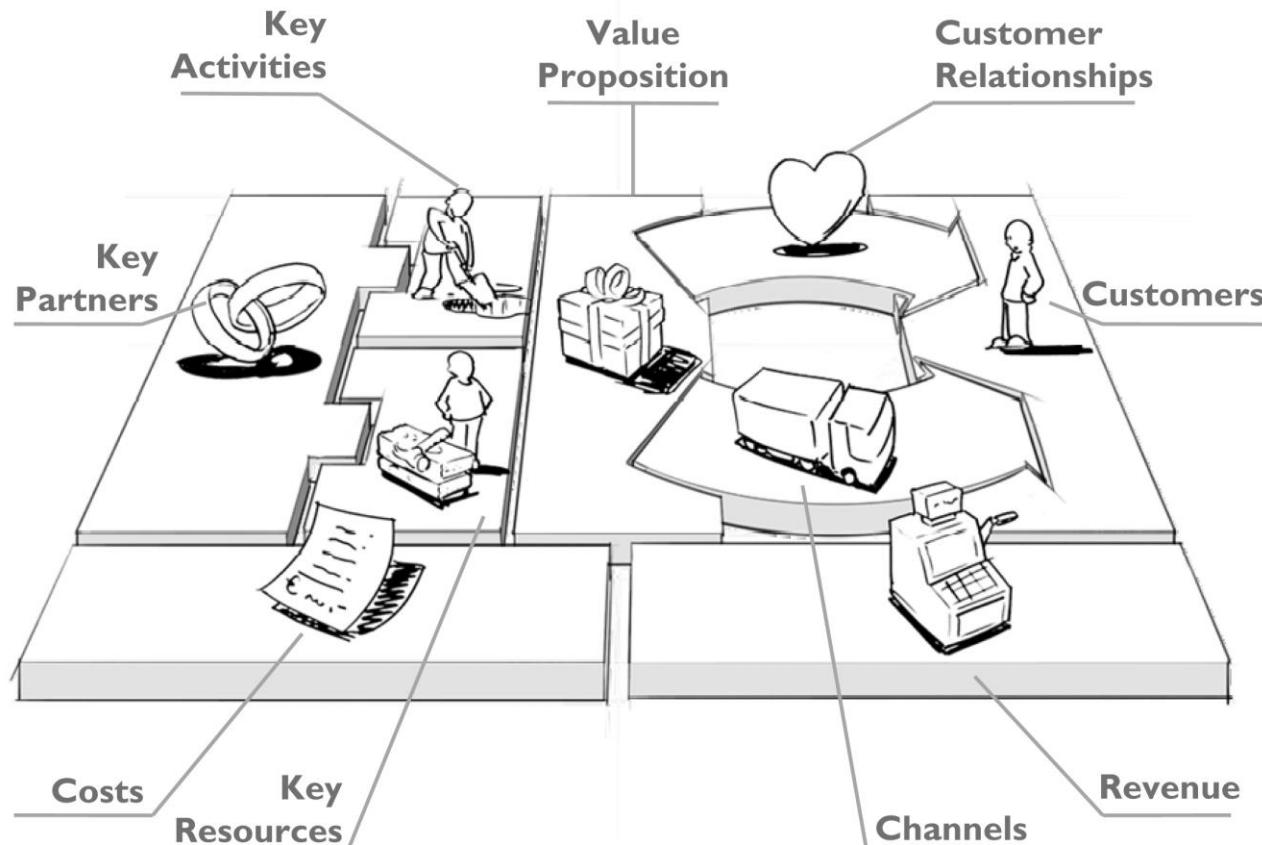


The Five Forces That Shape Industry Competition





Análisis Estratégico



drawings by JAM



El reto estratégico

- Valores
 - ¿Cuál es la misión?
 - ¿Cuál es el alcance del negocio?
 - ¿Cómo se genera valor?
- Oportunidades (¿Qué desea el mercado?)
- Capacidades (Ventaja Competitiva)



Ventaja Competitiva

No solo lo que se hace bien, sino lo que se hace mejor que otros para generar valor

“Si cualquiera puede hacerlo, es difícil crear y capturar valor”

(Principio de estrategia de negocio)



El reto estratégico

Se puede definir que estrategia son las acciones que toma una empresa para crear y generar valor y mantenerlo en el tiempo (ventaja competitiva)



El reto estratégico

If you don't have a competitive advantage, don't compete

Jack Welch – Former chairman and CEO of General Electric



Estrategia de Negocio





Ejemplos Misión



En el inmediato futuro, la Universidad Javeriana impulsará prioritariamente la investigación y la formación integral centrada en los currículos; fortalecerá su condición de universidad interdisciplinaria; y vigorizará su presencia en el país, contribuyendo especialmente a la solución de las problemáticas siguientes:

- **La crisis ética y la instrumentalización del ser humano.**
- **El poco aprecio de los valores de la nacionalidad y la falta de conciencia sobre la identidad cultural.**
- **La intolerancia y el desconocimiento de la pluralidad y la diversidad.**
- **La discriminación social y la concentración del poder económico y político.**
- **La inadecuación e inefficiencia de sus principales instituciones.**
- **La deficiencia y la lentitud en el desarrollo científico y tecnológico.**
- **La irracionalidad en el manejo del medio ambiente y de los recursos naturales.**



Ejemplos Misión



Google

La misión de Google es organizar la información del mundo y hacerla accesible y útil de forma universal



Ejemplos Misión



The Facebook logo is displayed in its signature blue color, with the word "facebook" in lowercase and a registered trademark symbol (®) at the end.

Founded in 2004, Facebook's mission is to give people the power to share and make the world more open and connected. People use Facebook to stay connected with friends and family, to discover what's going on in the world, and to share and express what matters to them.



Ejemplos Misión



La misión del Grupo Aval es proporcionar a nuestros clientes soluciones financieras socialmente responsables, seguras, fáciles de acceder, entender y manejar en cualquier lugar y hora que lo requieran, a través de los vehículos legales disponibles en cada uno de los mercados donde operamos; estas soluciones deben ser además rentables para nuestros clientes y para nuestras empresas y de esa manera redundar en generación de valor para nuestros accionistas.



¿Cómo se puede usar la misión de una compañía para entender su estrategia?

Escoger una opción:

- Se puede aprender de los valores de la organización, de lo que motiva e inspira a la compañía desde la misión, lo cual es clave para entender que está tratando de alcanzar y por qué.
- Debido a que las compañías son primariamente motivadas por las ganancias y para maximizar los retornos a los stakeholders, ver la misión no ayuda a entender la estrategia.



¿Cómo se puede usar la misión de una compañía para entender su estrategia?

Escoger una opción:

- **Se puede aprender de los valores de la organización, de lo que motiva e inspira a la compañía desde la misión, lo cual es clave para entender que está tratando de alcanzar y por qué.**
- **Debido a que las compañías son primariamente motivadas por las ganancias y para maximizar los retornos a los stakeholders, ver la misión no ayuda a entender la estrategia.**



¿Qué es lo que significa el principio de estrategia de negocios?

Escoger una opción:

- Si se está compitiendo en un mercado competitivo, entonces es muy probable que se vean ganancias.
- Solo cuando es difícil para otros hacer lo que una organización hace bien, entonces es muy probable que se vean ganancias.



¿Qué es lo que significa el principio de estrategia de negocios?

Escoger una opción:

- **Si se está compitiendo en un mercado competitivo, entonces es muy probable que se vean ganancias.**
- **Solo cuando es difícil para otros hacer lo que una organización hace bien, entonces es muy probable que se vean ganancias.**



Valor

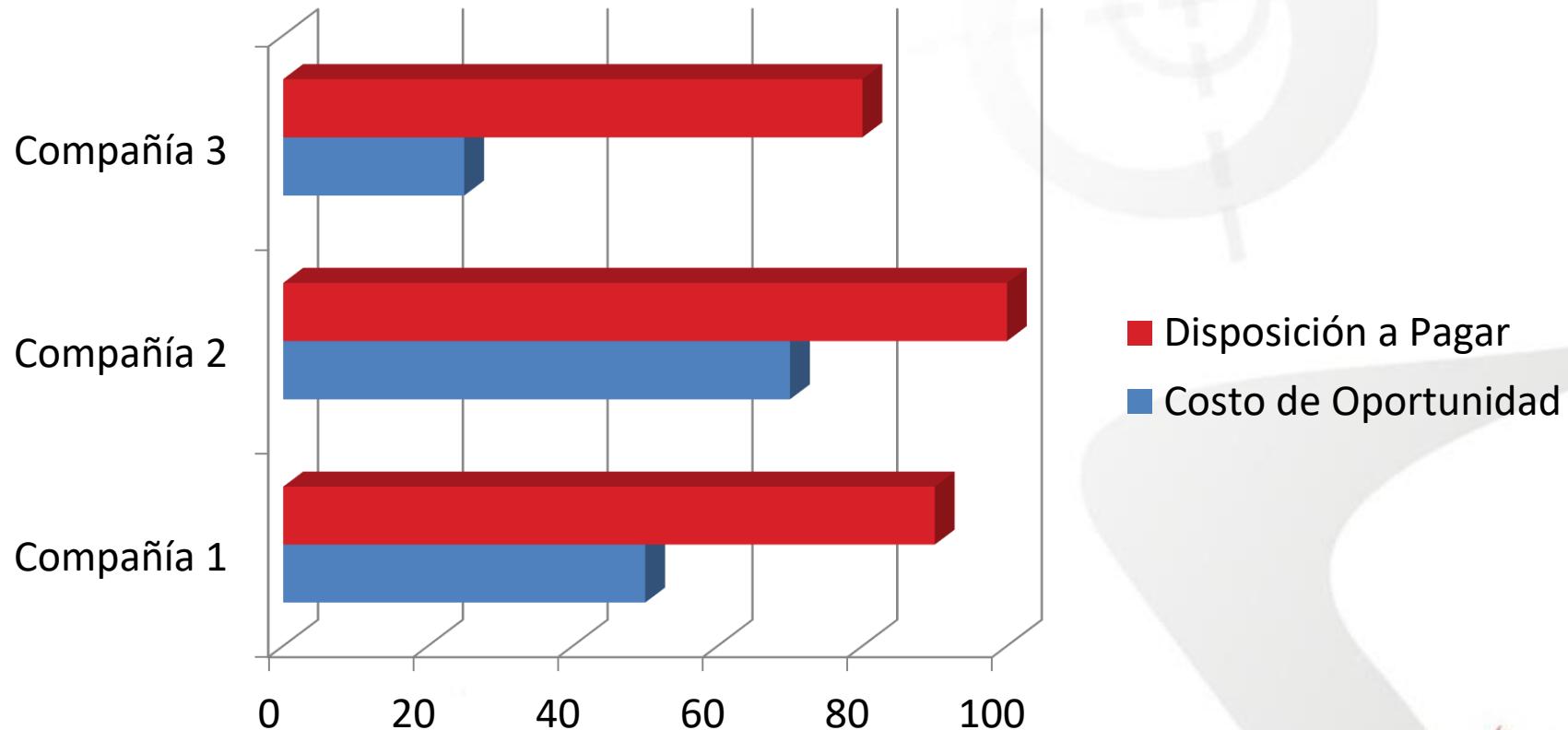


Valor

- **valor.**
- (Del lat. *valor, -ōris*).
- **1.** m. Grado de utilidad o aptitud de las cosas, para satisfacer las necesidades o proporcionar bienestar o deleite.
- **2.** m. Cualidad de las cosas, en virtud de la cual se da por poseerlas cierta suma de dinero o equivalente.
- **3.** m. Alcance de la significación o importancia de una cosa, acción, palabra o frase.



¿Qué es generar valor?





Pontificia Universidad
JAVERIANA
Bogotá

ESPECIALIZACIÓN EN
ARQUITECTURA EMPRESARIAL DE SOFTWARE

Modelos y Arquitectura



Modelo

modelo.

(Del it. *modello*).

1.m. **Arquetipo o punto de referencia para imitarlo o reproducirlo.**

2.m. En las obras de ingenio y en las acciones morales, ejemplar que por su perfección se debe seguir e imitar.

3. m. **Representación en pequeño de alguna cosa.**

4. m. Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para **facilitar su comprensión** y el estudio de su comportamiento.

5.m. Objeto, aparato, construcción, etc., o conjunto de ellos realizados con arreglo a un mismo diseño. *Auto modelo 1976. Lavadora último modelo.*

6.m. Vestido con características únicas, creado por determinado modista, y, en general, cualquier prenda de vestir que esté de moda.

7.m. En empresas, **u.** en aposición para indicar que lo designado por el nombre anterior ha sido creado como ejemplar o se considera que puede serlo. *Empresa modelo. Granjas modelo.*

8.m. **Esc.** Figura de barro, yeso o cera, que se ha de reproducir en madera, mármol o metal.



Modelo de un Sistema

- A model is a formal specification of the function, structure and behavior of a system within a given context, and from a specific point of view (or reference point). A model is often represented by a combination of drawings and text, typically using a formal notation such as UML, augmented where appropriate with natural language expressions.





Modelo de un Sistema

- A specification is said to be formal when it is based on a language that has a well defined semantic meaning associated with each of its constructs, to distinguish it from a simple diagram showing boxes and lines. It is this formalism, which allows the model to be expressed in a format such as XML, in accordance with a well-defined schema (XMI).





Modelo de un Sistema

- Representación:
 - Función
 - Estructura
 - Comportamiento
- Describen sistemas existentes
- Especifican sistemas para ser construidos
- Su lenguaje de representación es una combinación de dibujos y texto
- Un lenguaje formal tiene semántica y sintaxis. Adicionalmente puede tener reglas de análisis, inferencia o validación

Significado de los signos de un lenguaje y sus combinaciones

Conjunto de reglas que definen las secuencias correctas de los elementos de un lenguaje



Arquitectura



Real
Academia
Española

arquitectura.

(Del lat. *architectūra*).

1. f. Arte de proyectar y construir edificios.

2. f. Inform. Estructura lógica y física de los componentes de un computador.

~ civil.

1. f. Arte de construir edificios y monumentos públicos y particulares no religiosos.

~ hidráulica.

1. f. Arte de conducir y aprovechar las aguas, o de construir obras debajo de ellas.

~ militar.

1. f. Arte de fortificar.

~ naval.

1. f. Arte de construir embarcaciones.

~ religiosa.

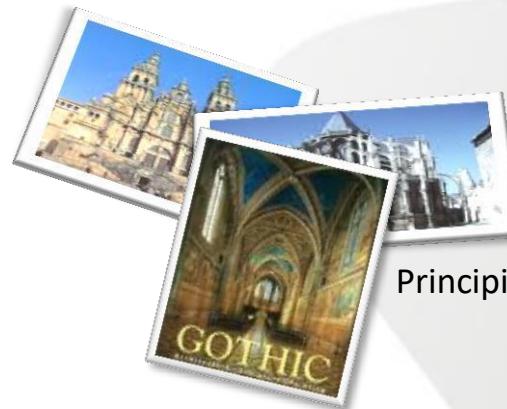
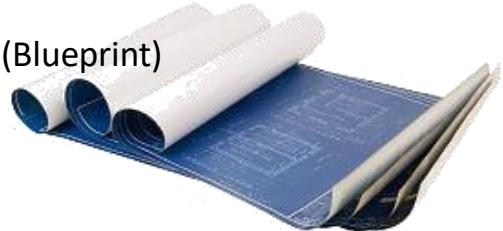
1. f. Arte de construir templos, monasterios, sepulcros y otras obras de carácter religioso.



Arquitectura

- Architecture is the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principle guiding its design and evolution.
(ISO/IEC 42010:2007)

Plano (Blueprint)



Principios (estilo)



Arquitectura

TOGAF

- A formal description of a system, or a detailed plan of the system at a component level to guide its implementation
- The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time



Arquitectura

A software architecture is an abstraction of the run-time elements of a software system during some phase of its operation. A system may be composed of many levels of abstraction and many phases of operation, each with its own software architecture.

A software architecture is defined by a configuration of architectural elements--components, connectors, and data--constrained in their relationships in order to achieve a desired set of architectural properties.

(Fielding, Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. 2000)

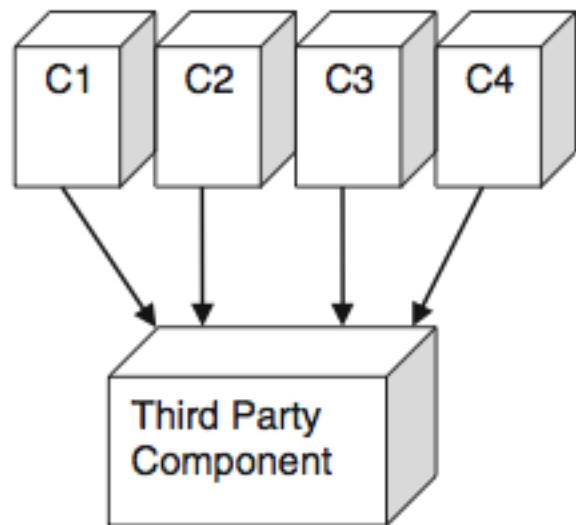


Arquitectura - Define la estructura de un sistema

- Software elements
 - Elements are capture as abstractions
 - Correspond to high level system modules or components
- External visible properties of elements
 - Describe element features exposed to others
 - Typically represent services provided to other elements
- Relationships of elements
 - Describe how elements interact with others

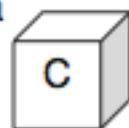


Arquitectura - Define la estructura de un sistema



Four components are directly dependent on a third party component. If the third party component is replaced with a new component with a different interface, changes to each component are likely.

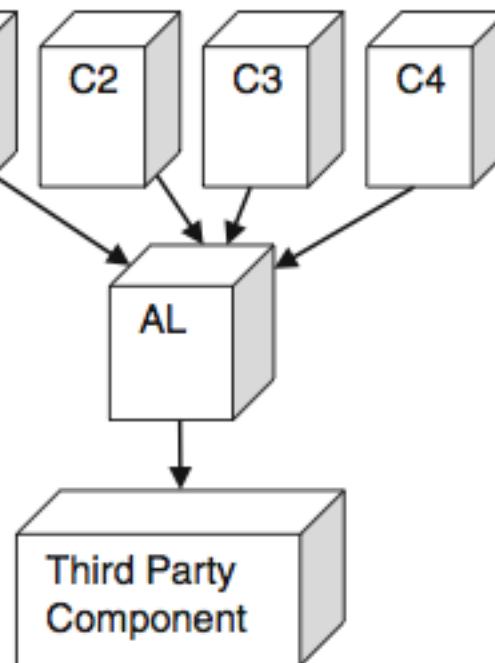
Diagram Key



Component



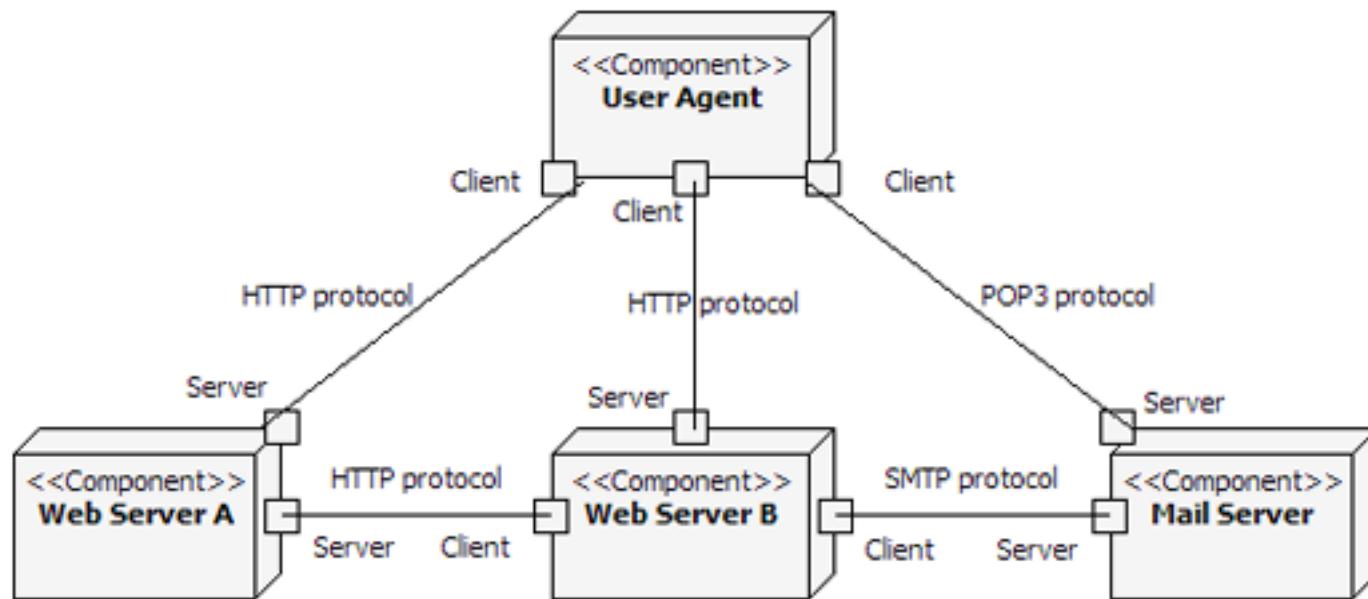
Dependency



Only the AL (abstraction layer) component is directly dependent on the third party component. If the third party component is replaced, changes are restricted to the AL component only.



Arquitectura - Define la comunicación entre componentes



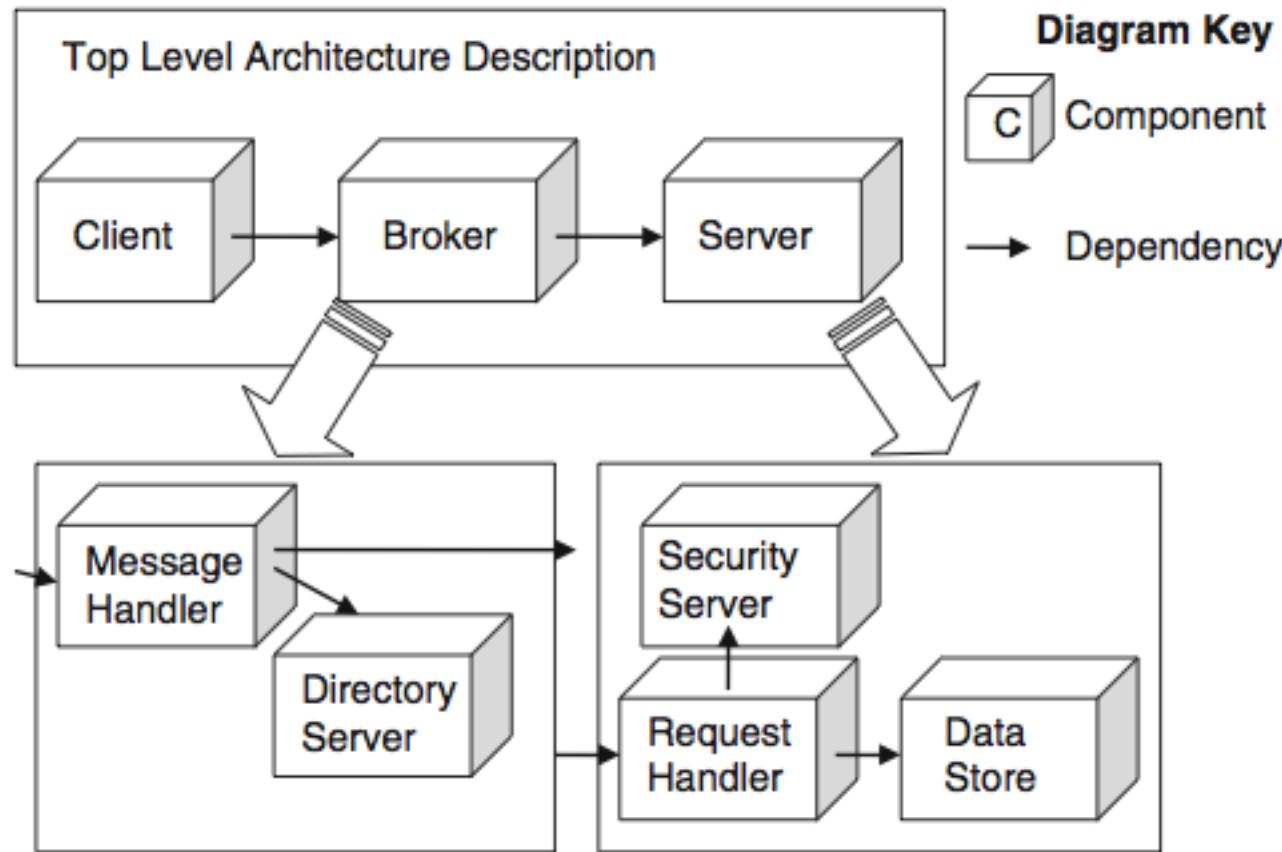


Arquitectura - La arquitectura trata los requerimientos no funcionales

- Restricciones técnicas
- Restricciones de negocio
- Atributos de calidad



Arquitectura como una abstracción





Empresa



Real
Academia
Española

empresa.

(Del it. *impresa*).

1.f. Acción o tarea que entraña dificultad y cuya ejecución requiere decisión y esfuerzo.

2.f. Unidad de organización dedicada a actividades industriales, mercantiles o de prestación de servicios con fines lucrativos.

3. f. Lugar en que se realizan estas actividades.

4. f. Intento o designio de hacer algo.

5.f. Símbolo o figura que alude a lo que se intenta conseguir o denota alguna prenda de la que se hace alarde, acompañada frecuentemente de una palabra o mote.

~ **pública.**

1. f. La creada y sostenida por un poder público.



Empresa

TOGAF

- A **collection of organizations** that share a common set of goals
 - Government agency
 - Part of a corporation
 - Corporation
- Large corporations may comprise multiple enterprises
- May be an “extended enterprise” including partners, suppliers and customers



Arquitectura Empresarial

- Enterprise architecture consists of defining and understanding the different elements that make up the enterprise, and how those elements are interrelated (**Open Group, 2003**).
- Enterprise architecture is the holistic expression of an organization's key business, information, application, and technology strategies and their impact on business functions and processes. The approach looks at business processes, the structure of the organization, and what type of technology is used to conduct these business processes (**Meta Group, 2005**).



Arquitectura Empresarial

- Enterprise architecture is a relatively simple and straightforward model, framework, or template that can be used by everyone within your enterprise to assess how things are going, to facilitate their work, and to design new projects (**Egan, 1988**).
- Enterprise architecture is the set of representations required to describe a system or enterprise regarding its construction, maintenance, and evolution (**Zachman, 1987**).
- Enterprise architecture is a strategic information asset base that defines the business mission, the information necessary to perform the mission, the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs (**FEAPMO, 2003**).



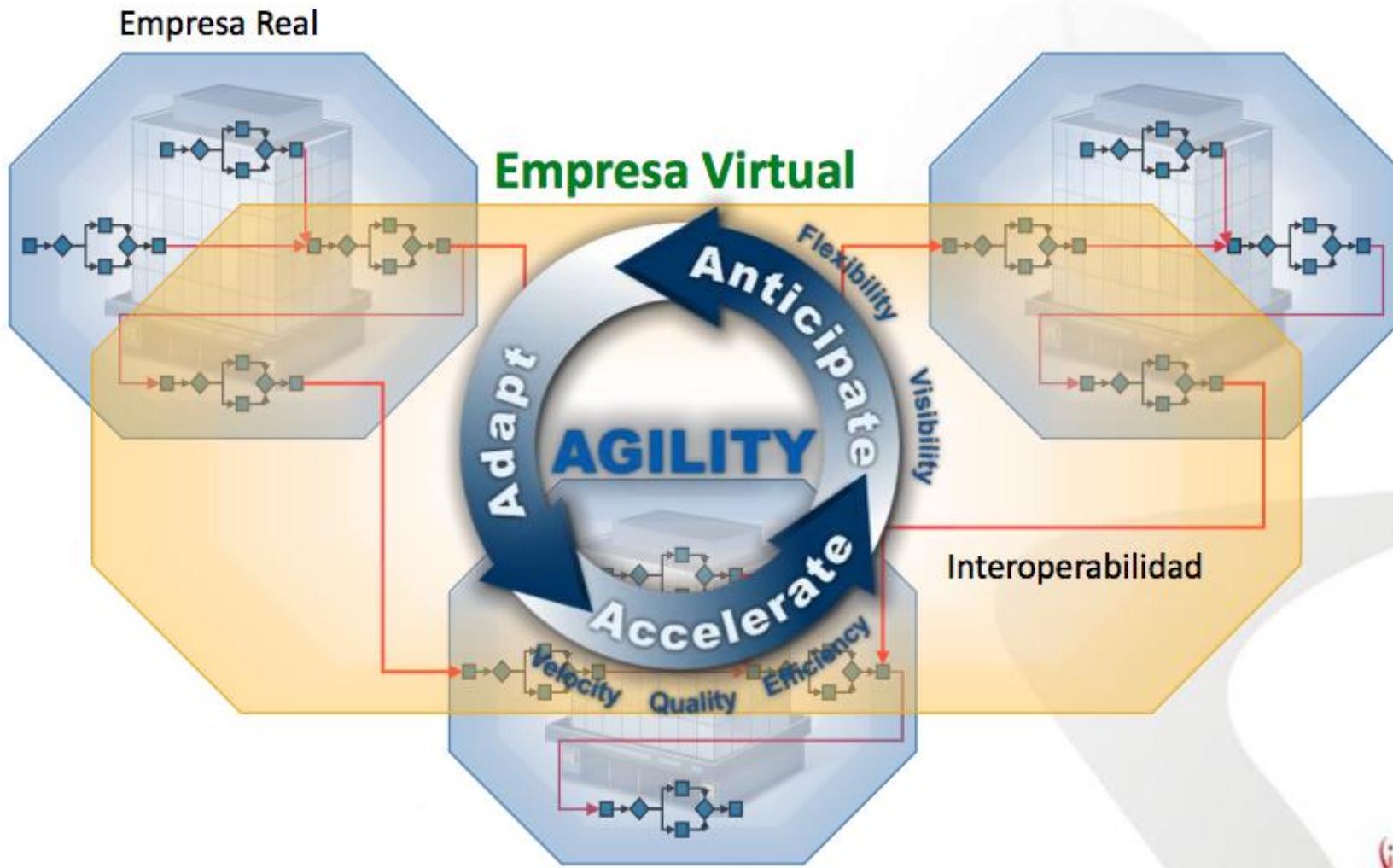
¿Qué busca una empresa?





Retos de las aplicaciones empresariales

- Incremento de la productividad del desarrollo
- Responder al incremento de la demanda
- Mantener el valor de los sistemas de información existentes
- Mantener la seguridad de los sistemas





Framework de Arquitectura

- Toolkit which can be used for developing a broad range of different architectures. (**Open Group, 2003**).
- Describe un método para diseñar un sistema de información en términos de bloques de construcción (**building blocks**) , y para mostrar como los bloques encajan entre sí.
- Contiene un conjunto de herramientas y provee un **vocabulario común**.
- Incluye una lista de recomendaciones de estándares y productos que pueden ser usados para implementar los bloques de construcción.



¿Por qué un Framework?

- Provee un punto de partida práctico para un proyecto de Arquitectura
 - Evita el pánico inicial con la aparente escala de la tarea a realizar
 - Sistemático – Sentido común
 - Recopila lo que otros han encontrado en la vida real
 - Contiene una línea base de recursos para reuso



Framework de Zachman de Arquitectura Empresarial

	WHAT	HOW	WHERE	WHO	WHEN	WHY
SCOPE (contextual)	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
Planner	List of Things Important to the Business Entity = Class of Business Thing	List of Processes the Business Performs Process = Class of Business Process	List of Locations in Which the Business Operates Node = Major Business Location	List of Organizations Important to the Business People = Major Organisational Unit	List of Events/Cycles Significant to the Business Time = Major Business Event/Cycle	Lists of Business Goals/Strategies Ends/Mean = Major Business Goal/Strategy
Owner	e.g., Semantic Model Entity = Business Entity Relationship = Business Relationship	e.g., Business Process Model Process = Business Process I/O = Business Resources	e.g., Business Logistics System Node = Business Location Link = Business Linkage	e.g., Work Flow Model People = Organization Unit Work = Work Product	e.g., Master Schedule Time = Business Event Cycle = Business Cycle	e.g., Business Plan End = Business Objective Means = Business Strategy
Designer	e.g., Logical Data Model Entity = Data Entity Relationship = Data Relationship	e.g., Application Architecture Process = Application Function I/O = User Views	e.g., Distributed System Architecture Node = I/O Function (Processor, Storage, etc.) Link = Line Characteristics	e.g., Human Interface Architecture People = Role Work = Deliverable	e.g., Processing Structure Time = System Event Cycle = Processing Cycle	e.g., Business Rule Model End = Structural Assertion Means = Action Assertion
Builder	e.g., Physical Data Model Entity = Segment/Table/etc. Relationship = Pointer/Key/etc.	e.g., System Design Process = Computer Function I/O = Data Events/Sets	e.g., Technology Architecture Node = HW/System Software Link = Line Specifications	e.g., Presentation Architecture People = User Work = Screen Formats	e.g., Control Structure Time = Execute Cycle = Component Cycle	e.g., Rule Design End = Condition Means = Action
Subcontractor	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language Statement I/O = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Security Architecture People = Identity Work = Job	e.g., Timing Definition Time = Interrupt Cycle = Machine Cycle	e.g., Rule Specification End = Sub-condition Means = Step
	FUNCTION	DATA	NETWORK	ORGANIZATION	SCHEDULE	STRATEGY
	DATA	FUNCTION	NETWORK	ORGANIZATION	SCHEDULE	STRATEGY



Alineación

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (contextual) Planner	List of Things Important to the Business 	List of Processes the Business Performs 	List of Locations in Which the Business Operates 	List of Organizations Important to the Business 	List of Events/Cycles Significant to the Business 	Lists of Business Goals/Strategies
BUSINESS MODEL (conceptual) Owner	e.g., Semantic Model 	e.g., Business Process Model 	e.g., Business Logistics System 	e.g., Work Flow Model 	e.g., Master Schedule 	e.g., Business Plan
SYSTEM MODEL (logical) Designer	e.g., Logical Data Model 	e.g., Application Architecture 	e.g., Distributed System Architecture 	e.g., Human Interface Architecture 	e.g., Processing Structure 	e.g., Business Rule Model
TECHNOLOGY MODEL (physical) Builder	e.g., Physical Data Model 	e.g., System Design 	e.g., Technology Architecture 	e.g., Presentation Architecture 	e.g., Control Structure 	e.g., Rule Design
DETAILED REPRESENTATIONS (out-of-context) Subcontractor	e.g., Data Definition 	e.g., Program 	e.g., Network Architecture 	e.g., Security Architecture 	e.g., Timing Definition 	e.g., Rule Specification



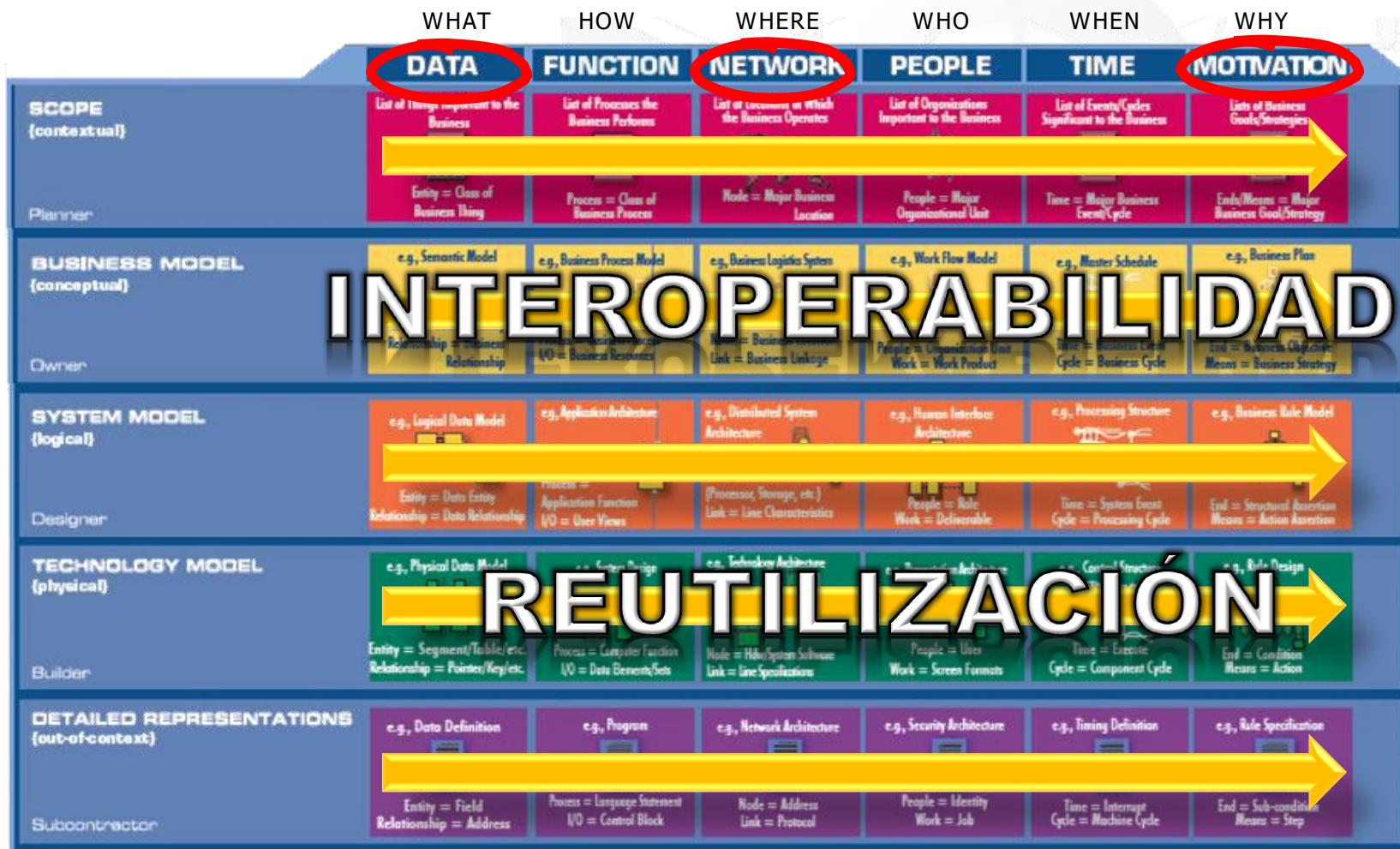
Alineación

	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
SCOPE (contextual)	List of Things Belonging to the Business	List of Business Processes	List of Locations of the Business	List of Organizational Units Important for business	List of Events Significant for business	List of Goals Business Strategies
Planner	Entity = Business Relationship = Category	Process = Business Process I/O = Business I/O	Node = Business Location Link = Business Connection	People = Organization Unit Work = Work Product	Time = Work Cycle Event = Business Event	End = Business End Means = Business Means
BUSINESS MODEL (conceptual)	e.g., Semantic Model Entity = Business Relationship	e.g., Business Model Process = Business I/O	e.g., Business Model Node = Business Location Link = Business Connection	e.g., Work Model People = Organization Unit Work = Work Product	e.g., Work Model Time = Work Cycle Event = Business Event	e.g., Business Plan End = Business End Means = Business Means
Owner						
SYSTEM MODEL (logical)	e.g., Logical Model Entity = Relationship	e.g., Application Model Entity = Application I/O = User	e.g., Distributed System Architecture Node = Host Link = Line	e.g., Human Resource Model Work = Work Available	e.g., Production Model Time = Work Cycle Event = Business Event	e.g., Business Model End = Business End Means = Business Means
Designer						
TECHNOLOGY MODEL (physical)	e.g., Physical Model Entity = Segmentable/etc. Relationship = Key/etc.	e.g., System Design Process = Function I/O = Interface	e.g., Technology Model Node = Hardware Link = Line	e.g., Processor Architecture Work = User Formats	e.g., Computer Architecture Time = Work Cycle Event = Business Event	e.g., System Design End = Business End Means = Business Means
Builder						
DETAILED REPRESENTATIONS (out-of-context)	e.g., Data Definition Entity = Field Relationship = Address	e.g., Statement Process = Language Statement I/O = Control Block	e.g., Network Architecture Node = Address Link = Protocol	e.g., Structure People = Identity Work = Job	e.g., Timetable Time = Interrupt Cycle = Machine Cycle	e.g., Specification End = Sub-condition Means = Step
Subcontractor						

VALIDACIÓN

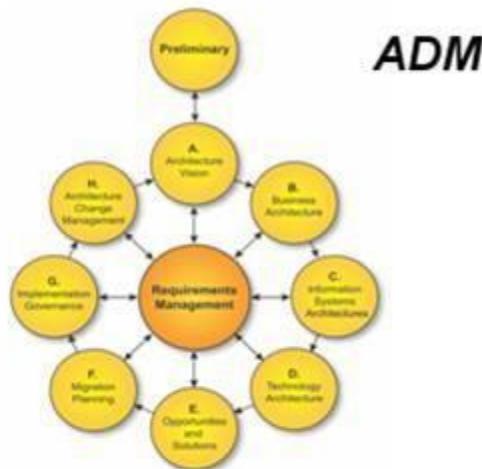


Integración





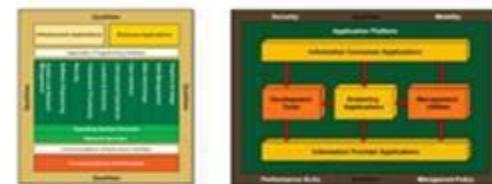
The Open Group Architecture Framework



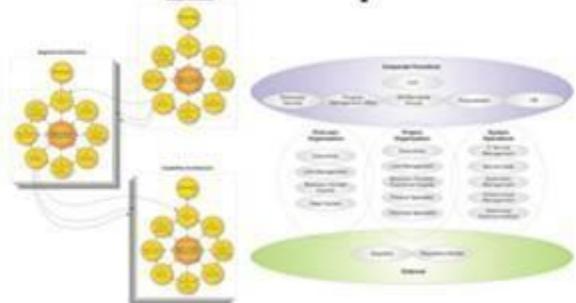
Architecture Content Framework



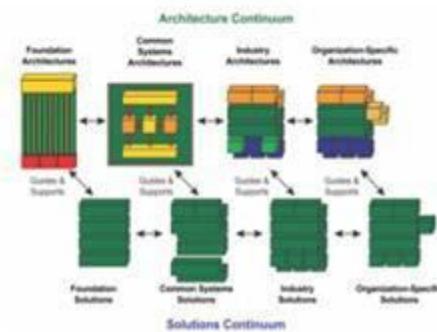
Reference Models



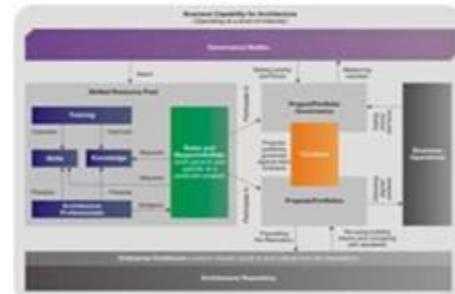
ADM Guidelines & Techniques



Enterprise Continuum

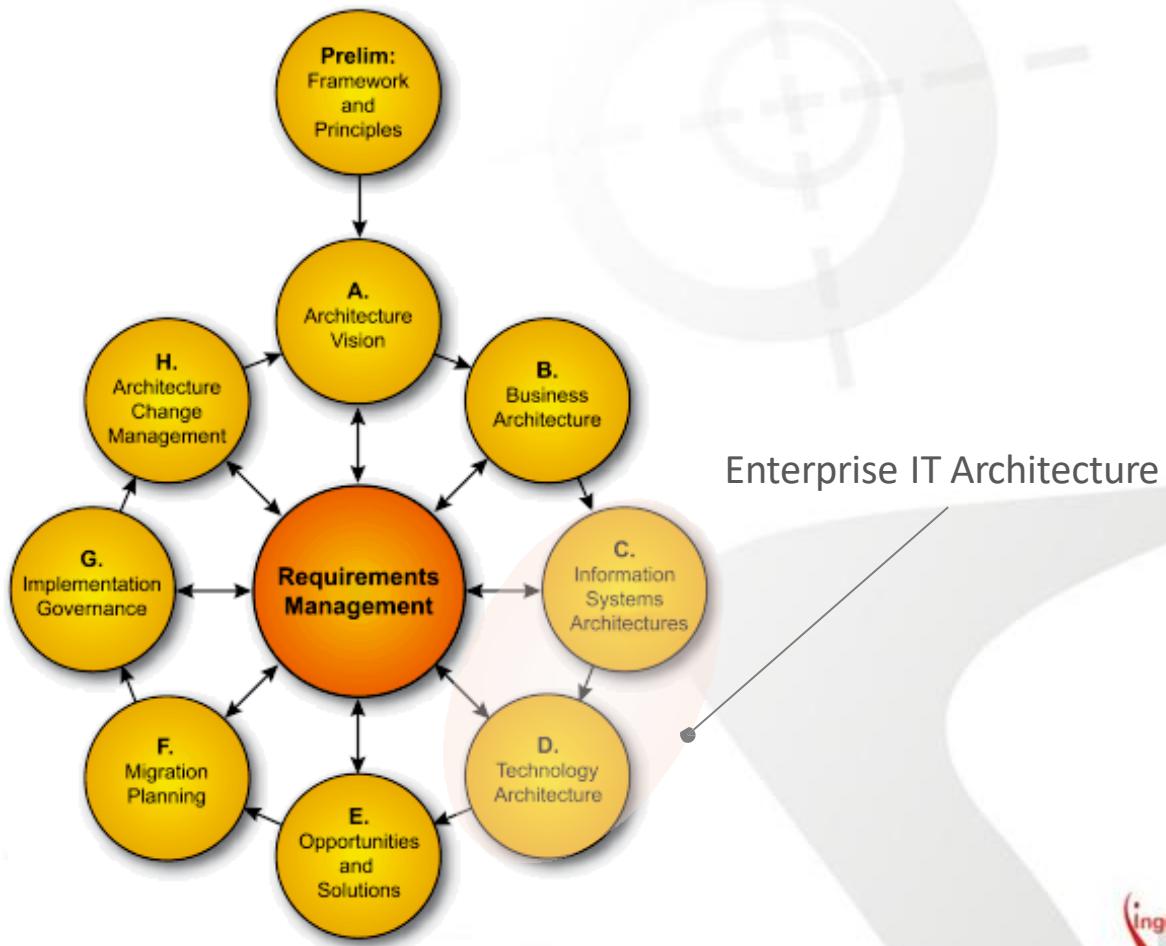


Architecture Capability Framework



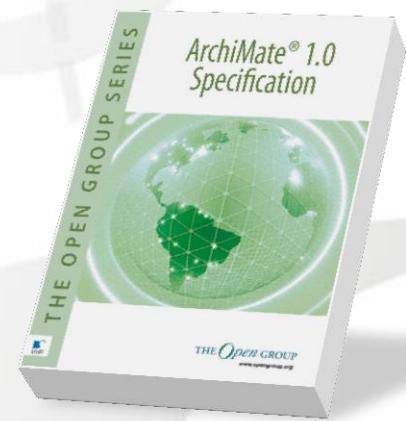
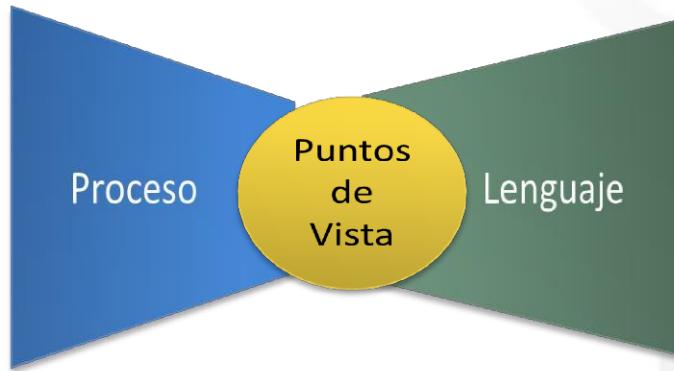
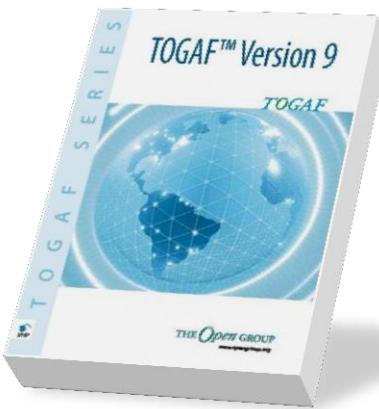


The Open Group Architecture Framework





Ingredientes de la Arquitectura Empresarial





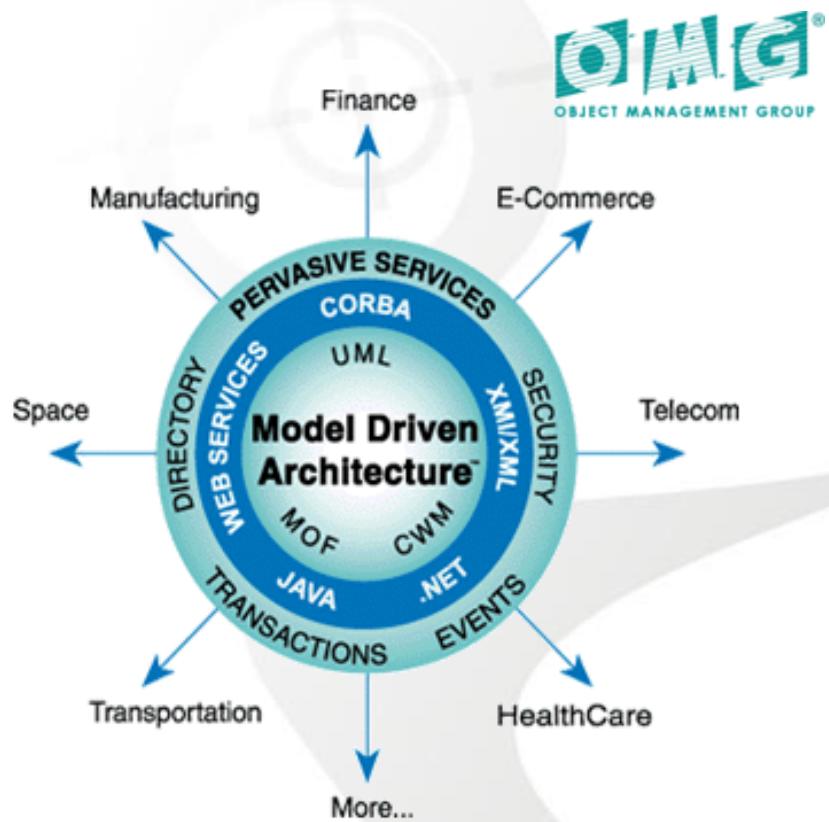
Model-Driven Architecture

- **Model**
 - A model is a formal specification of the function, structure and behavior of a system within a given context, and from a specific point of view (or reference point). A model is often represented by a combination of drawings and text, typically using a formal notation such as UML, augmented where appropriate with natural language expressions.
- **Model driven**
 - Describes an approach to software development whereby models are used as the primary source for documenting, analyzing, designing, constructing, deploying and maintaining a system.



Model Driven Architecture

- Objetivos
 - Portabilidad
 - Interoperabilidad
 - Reutilización
- Tareas
 - Especificar un sistema independiente de la plataforma
 - Especificar plataformas
 - Escoger una plataforma
 - Transformar la especificación independiente de la plataforma para una la plataforma escogida





Model-Driven Architecture

Viewpoints

- **Computation Independent:** focuses on the context and requirements of the system without consideration for its structure or processing.
- **Platform independent:** focuses on the operational capabilities of a system outside the context of a specific platform (or set of platforms) by showing only those parts of a complete specification that can be abstracted out of that platform.
- **Platform specific:** augments a platform independent viewpoint with details relating to the use of a specific platform.

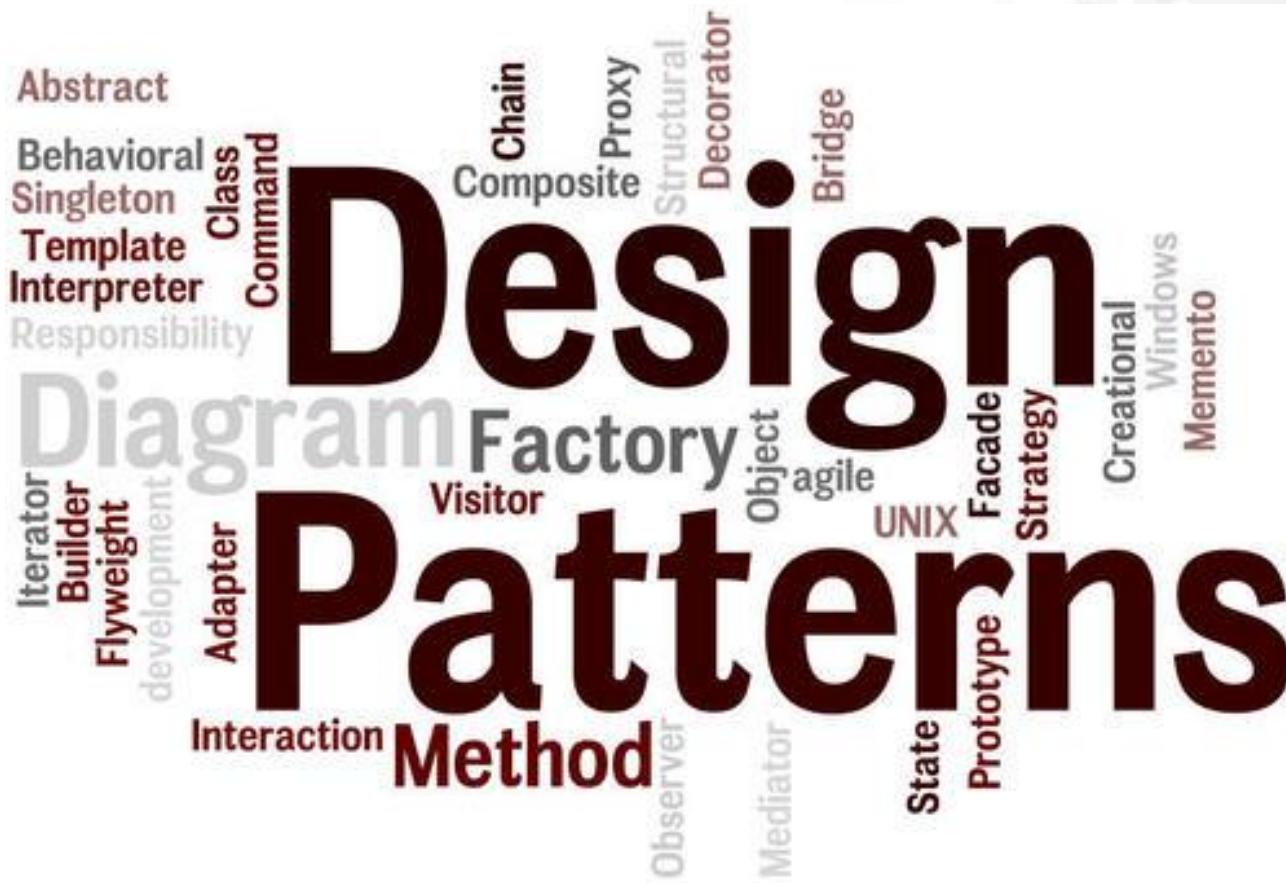


CIM: Computational Independent Model

- Expresa los requerimientos del sistema
- Describe la situación en la cual el sistema será usado
- Independiente de la forma como el sistema será implementado
- No tiene detalles de estructura o procesamiento del sistema
- Conocido como modelo de dominio o modelo de negocio



CIM: Computational Independent Model





PIM: Platform Independent Model

- Expresa la solución de manera abstracta
- Muestra las estructuras concretas y los mecanismos usados en la implementación
- No se ata a ninguna plataforma

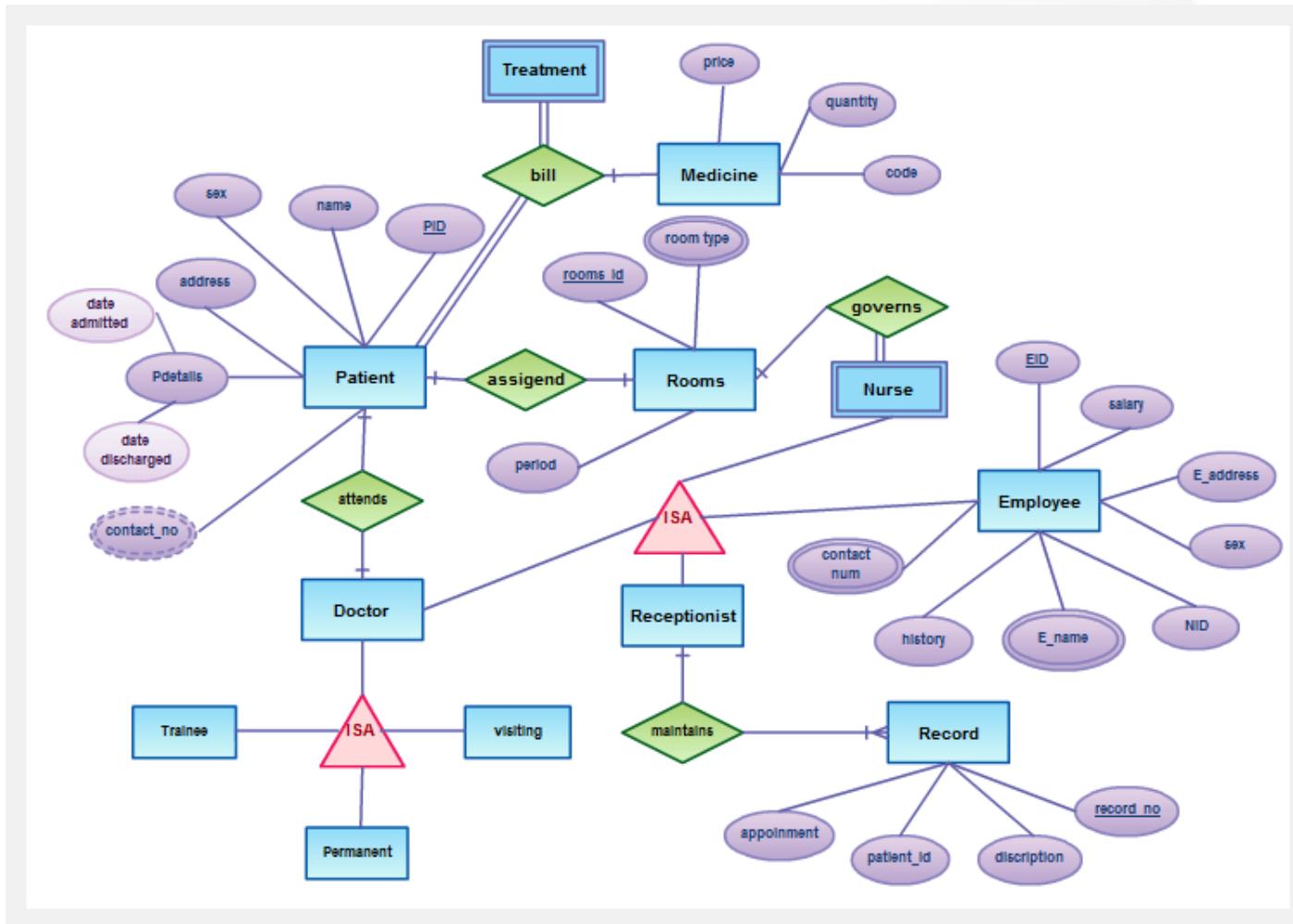


PSM: Platform Specific Model

- Modelos concretos con las estructuras y mecanismos correspondientes a un plataforma específica
- **Platform:** A set of subsystems/technologies that provide a coherent set of functionality through interfaces and specified usage patterns that any subsystem that depends on the platform can use without concern for the details of how the functionality provided by the platform is implemented (**OMG, 2003**)

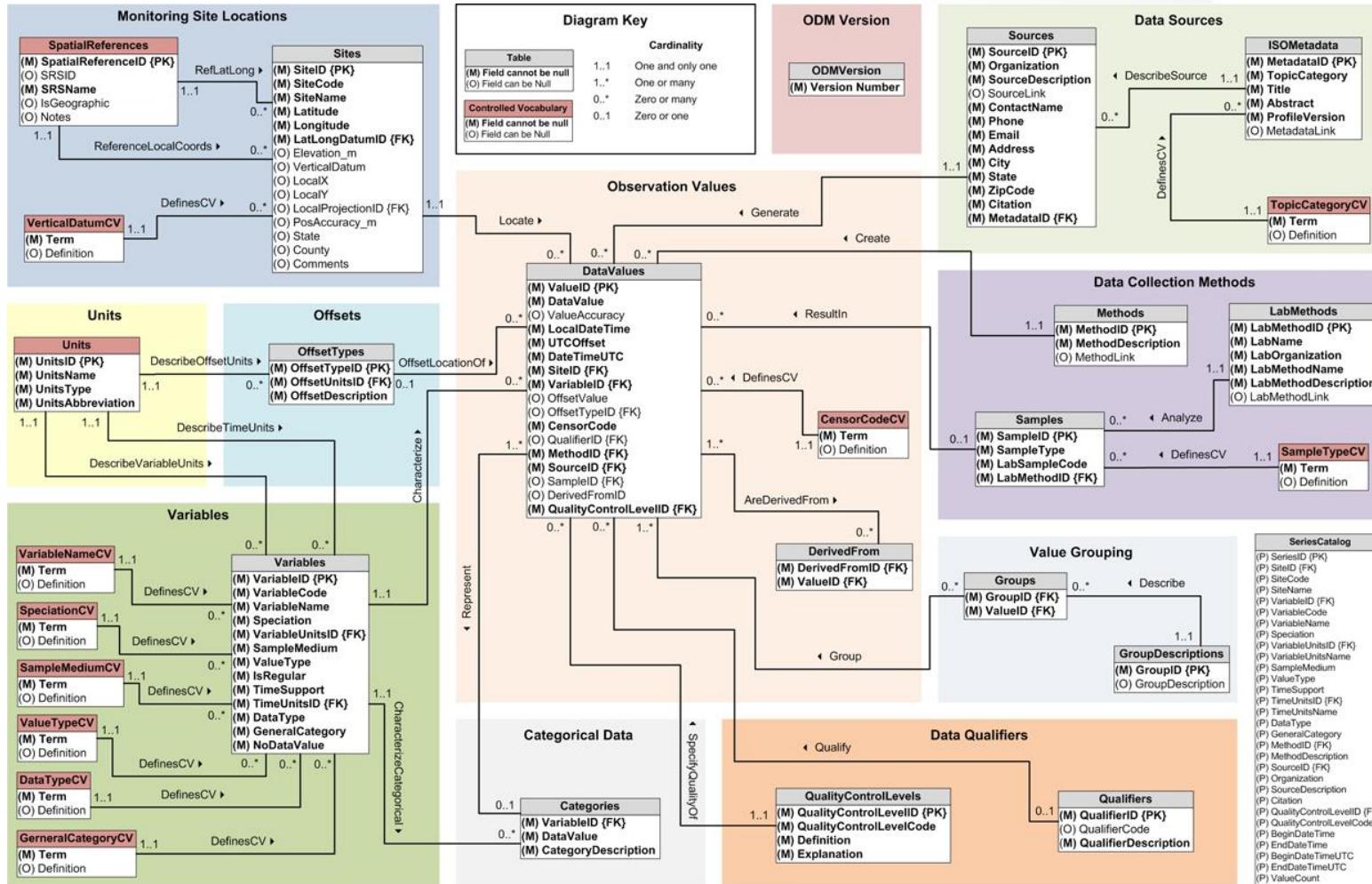


Model Driven Architecture - CIM





Model Driven Architecture - PIM





Model Driven Architecture - PSM

The screenshot shows the dbForge SQL Decryptor interface. On the left, the Object Explorer pane displays a tree structure of database objects for the 'AdventureWorks2008' database, specifically listing various DML Triggers such as dEmployee, dVendor, iduSalesOrderDetail, etc. In the center-right, a large text area contains a SQL script for a trigger named 'dEmployee'. The script uses the 'RAISERROR' command to prevent employees from being deleted, instead marking them as inactive. It also handles transactions to ensure consistency.

```
1 USE AdventureWorks2008
2 GO
3 SET ANSI_NULLS, QUOTED_IDENTIFIER ON
4 GO
5
6 CREATE TRIGGER [HumanResources].[dEmployee] ON [HumanResources].[Employee]
7 INSTEAD OF DELETE NOT FOR REPLICATION AS
8 BEGIN
9     DECLARE @Count int;
10
11    SET @Count = @@ROWCOUNT;
12    IF @Count = 0
13        RETURN;
14
15    SET NOCOUNT ON;
16
17    BEGIN
18        RAISERROR
19            (N'Employees cannot be deleted. They can only be marked
20              10, -- Severity.
21              1); -- State.
22
23        -- Rollback any active or uncommittable transactions
24        IF @@TRANCOUNT > 0
25            BEGIN
26                ROLLBACK TRANSACTION;
27            END
28        END;
29    END;
30    GO
31 DISABLE TRIGGER dEmployee ON HumanResources.Employee
32 GO
```

Model Driven Architecture - CIM

BPMN 2.0 - Business Process Model and Notation

<http://bpmb.de/poster>

Activities

- Task
- Exclusive OR Gate
- Event
- Call Activity
- Parallel Activities
- Activity Diagram
- Task Types:
 - Simple Task
 - Complex Task
 - Parallel Task
 - Event-based Task
 - Resource Pool Task
 - Manual Task
 - Script Task
- Response Flow:
 - Starts the associated object automatically.
 - Is the default flow to be executed if no other conditions evaluate to true.
 - Condition or Else:
 - Has a condition that must be met for the flow to occur.
 - Has a condition that must not be met for the flow to occur.

Conversations

- A Conversation defines a set of logically related messages between two participants. It consists of a participant and a corresponding conversation element.
- A Conversation is composed of Conversations and Participants.
- A Formal Conversation Link connects multiple conversations.

Choreographies

Participants:

- Choreography Task
- Process
- Particular Task
- Particular Process

Choreography Diagram

Collaboration Diagram

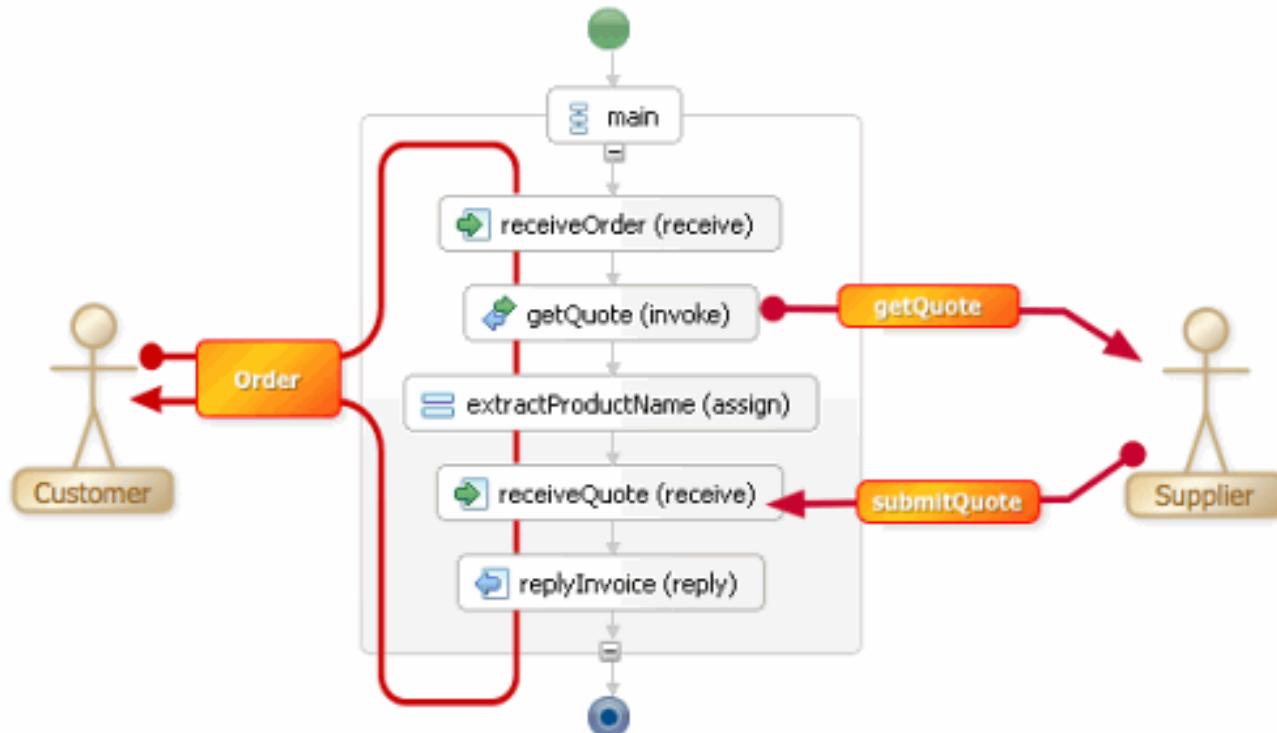
Events

Data

Swimlanes



Model Driven Architecture - PIM



Legend



WSDL/XML based service invocations



Service operations published by BPEL process



Model Driven Architecture - PSM

Screenshot of Oracle JDeveloper showing a BPEL process named "myCreditFlow.bpel".

The process diagram shows:

- A "client" partner link connected to a "receiveInput" activity.
- The "receiveInput" activity has a outgoing arrow labeled "main" leading to an "invoke" activity.
- The "invoke" activity has a outgoing arrow labeled "callbackClient" leading back to the "client" partner link.

The "Component Palette" on the right lists Process Activities: Pointer, Assign, Compensate, and Empty. The "receiveInput" activity is selected, showing its properties in the "Property Inspector":

createInstance	True
joinCondition	
name	receiveInput
operation	initiate
partnerLink	client
variable	inputVariable

The "Diagram View" shows a "Messages" tab with the message: "Validating Process... 0 errors and 0 warnings were found. Done validating."

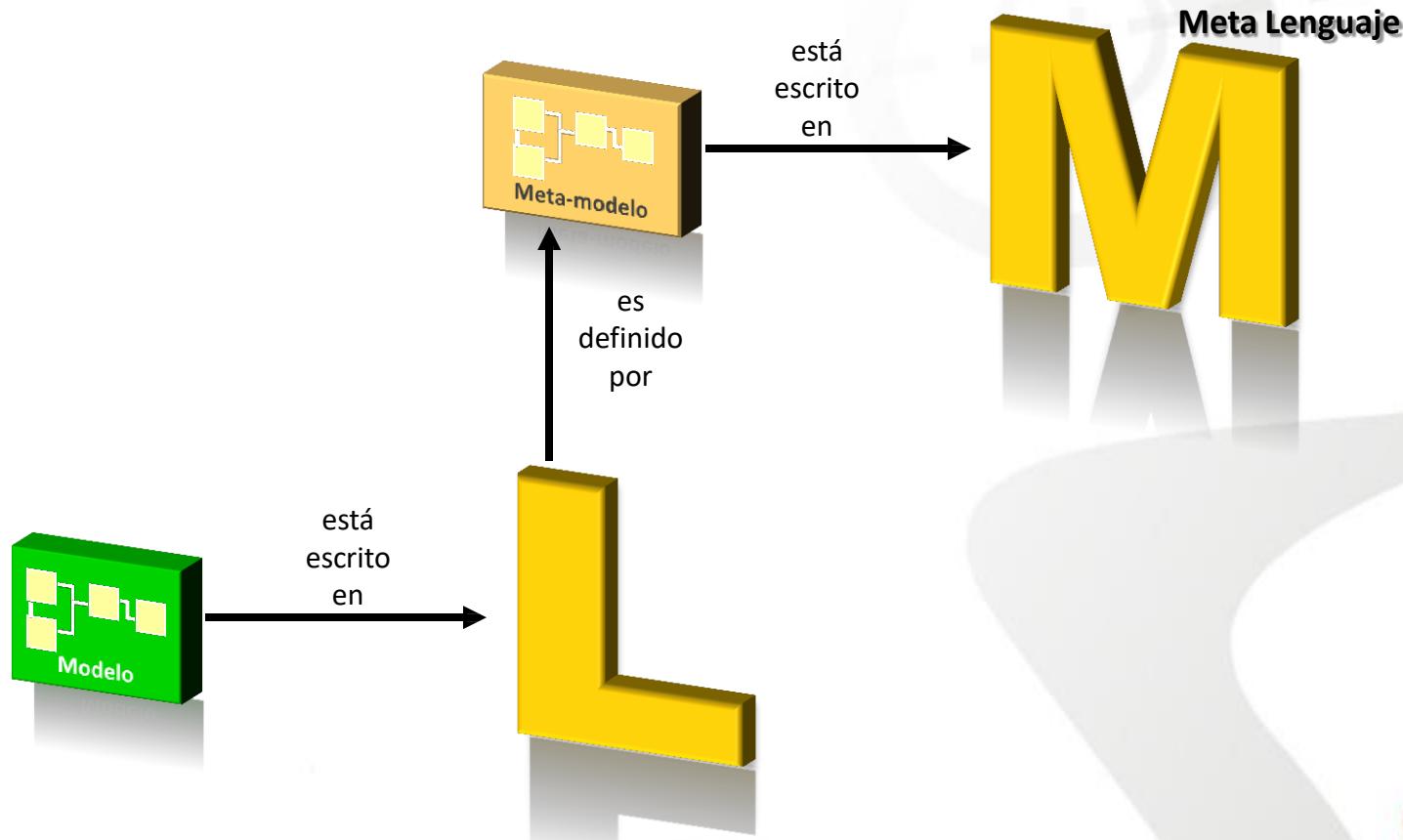


Meta-modelos

- Un modelo es una especificación formal de la función, estructura y/o comportamiento de una aplicación o sistema (**Object Management Group, 2003**)
- Modelo de Modelos (**Object Management Group, 2003**)
- Una manera estándar para definir la interoperabilidad entre modelos
- UML Profiles (estereotipos): Construir modelos en dominios particulares

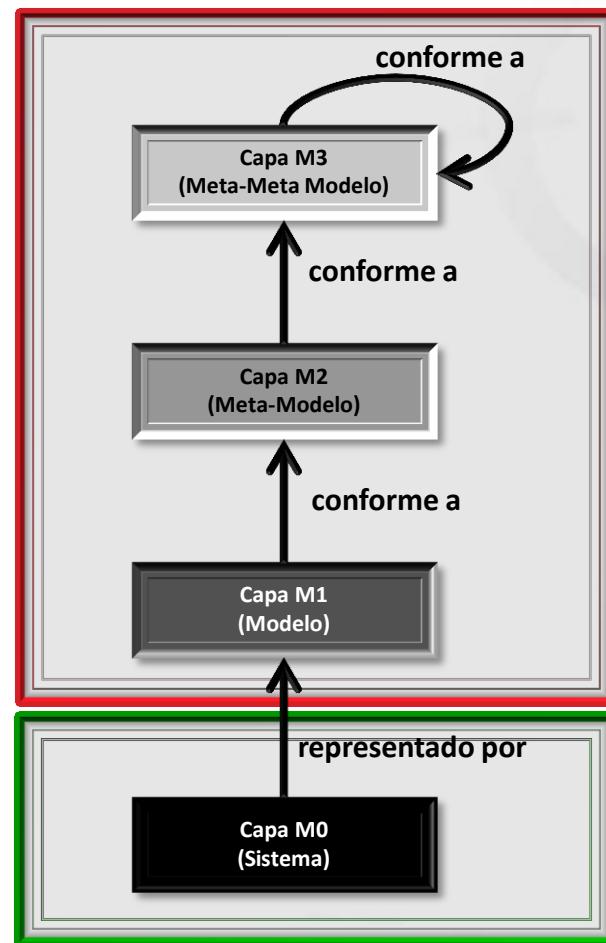


Meta-modelos y Lenguajes





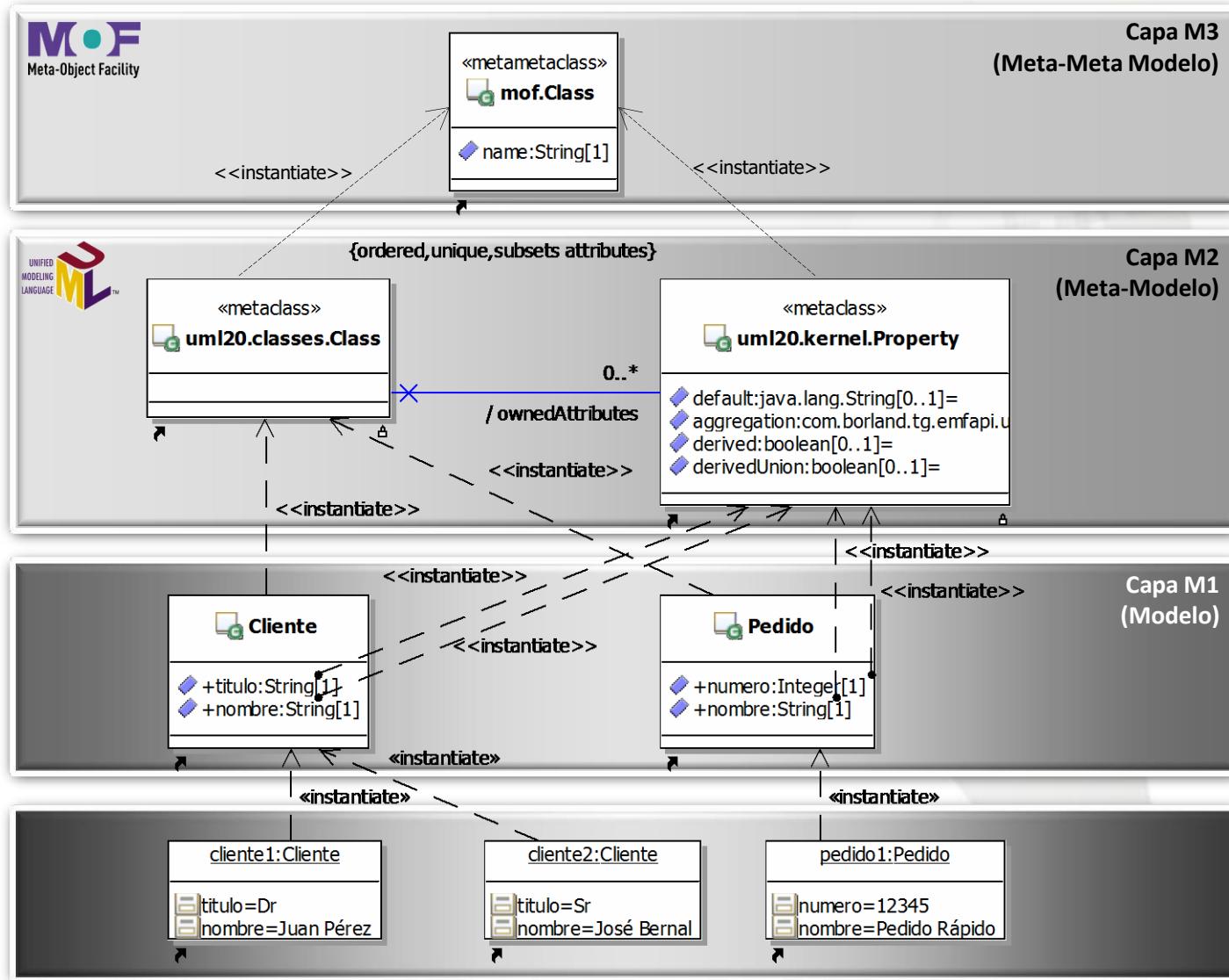
Capas de Modelado



Modelado
del
Mundo

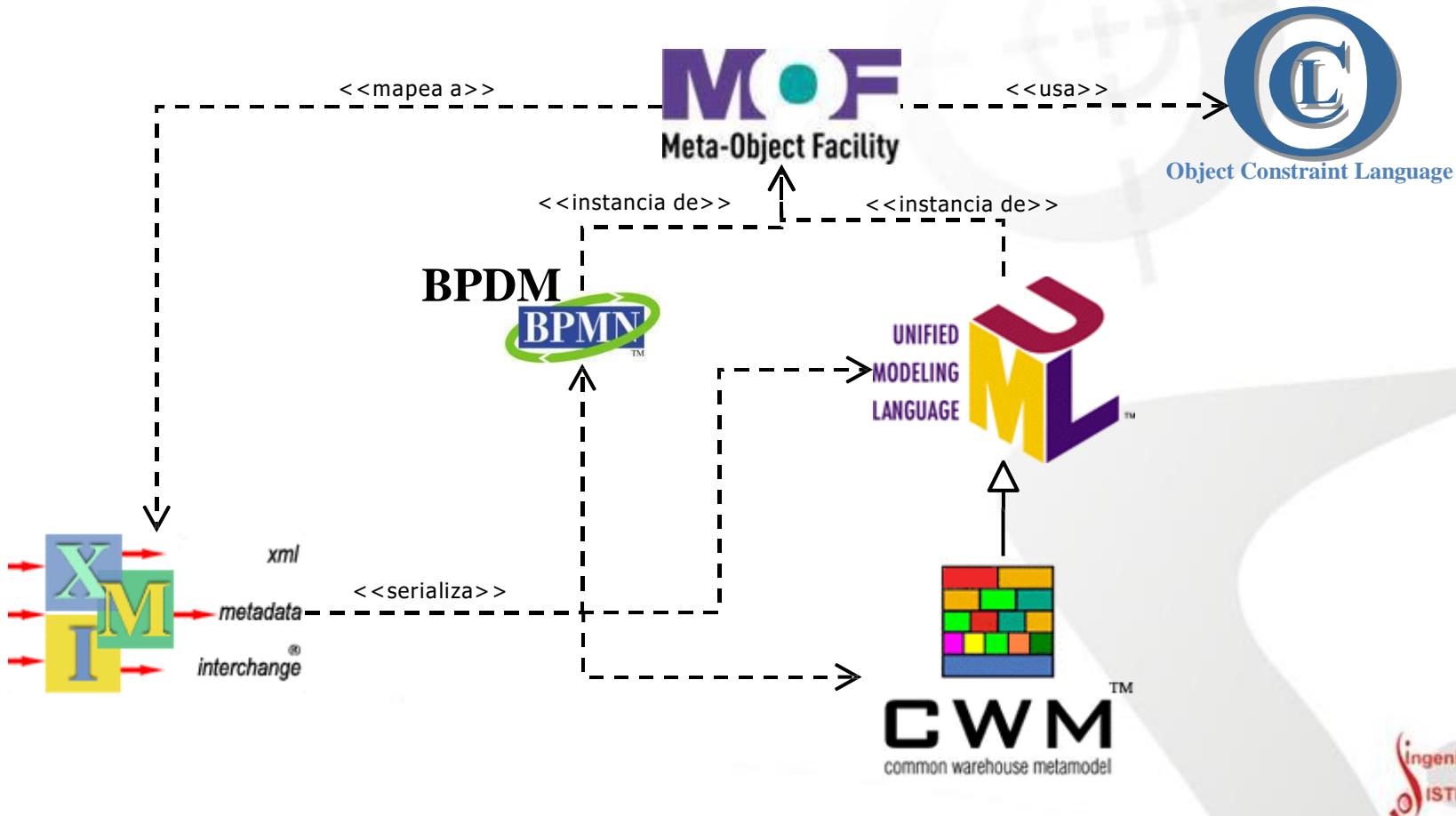
Mundo
Real







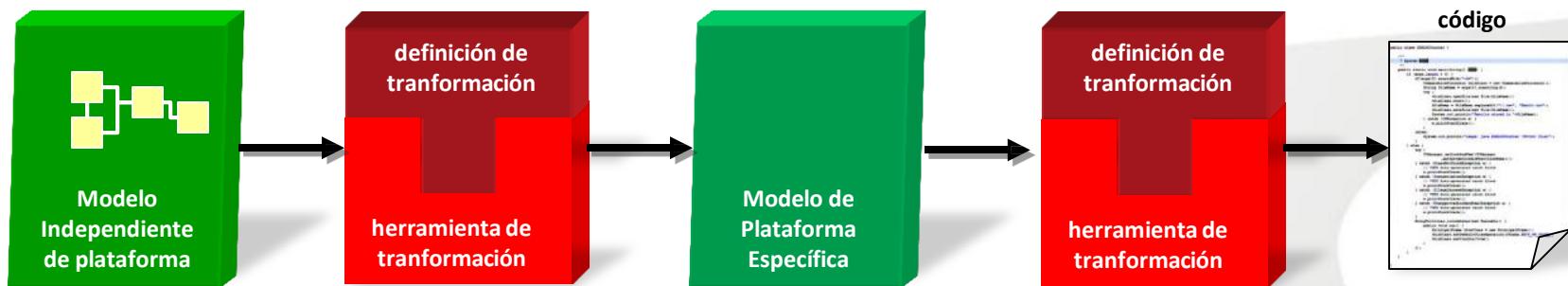
Otros estándares de OMG





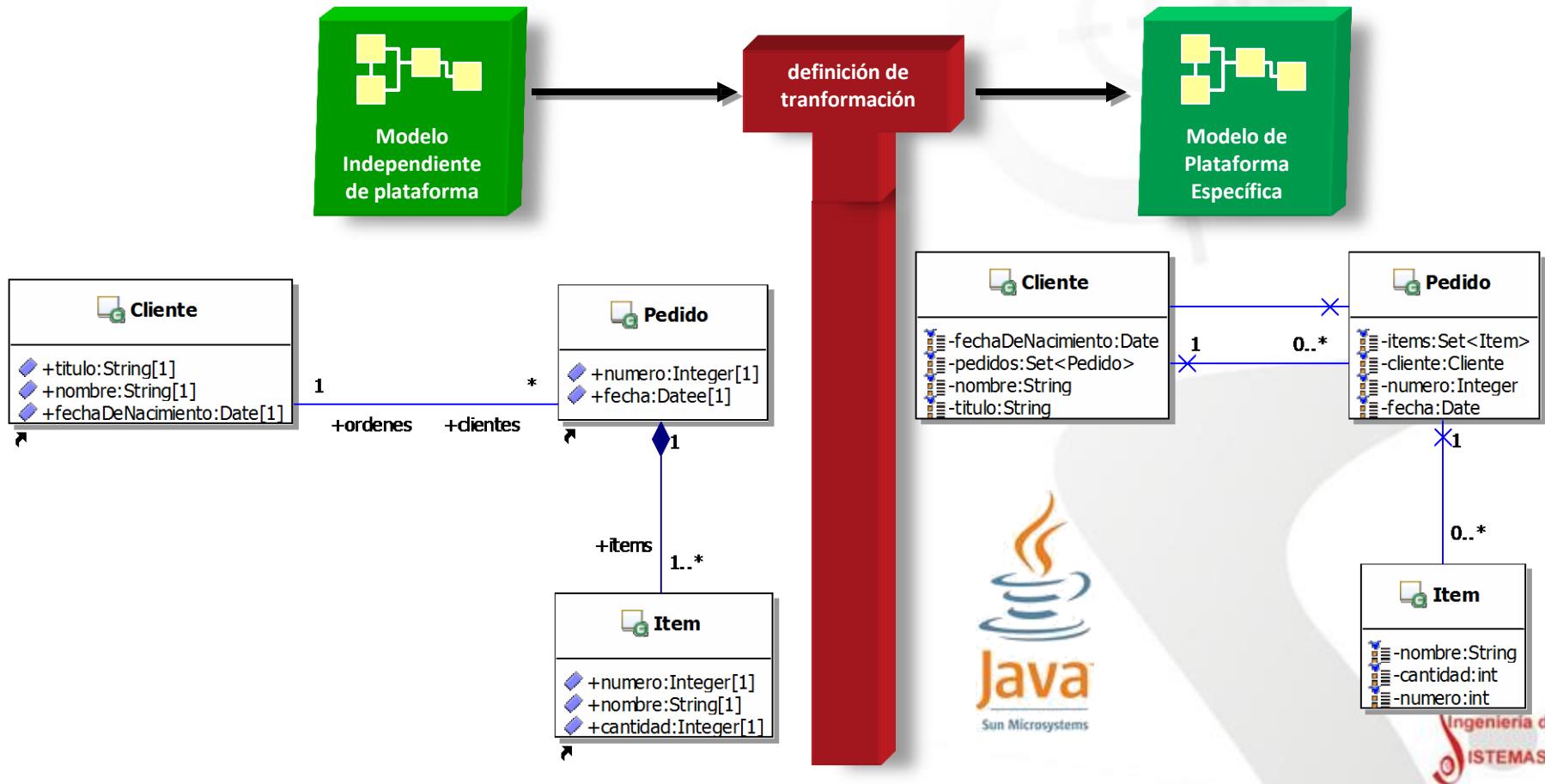
Transformación de Modelos

- El proceso de convertir un modelo a otro modelo del mismo sistema
(Object Management Group, 2003)





Ejemplo de Transformación de Modelos





Domain-Driven Design

“A useful package of software cannot be decoupled from that sphere of reality, the domain it is supposed to help us manage. On the contrary, the software is deeply entangled with it.”

Abel Avram & Floyd Marinescu
Domain-Driven Design Quickly



Domain-Driven Design



Volvo XC40
Concept Car vs Actual Car



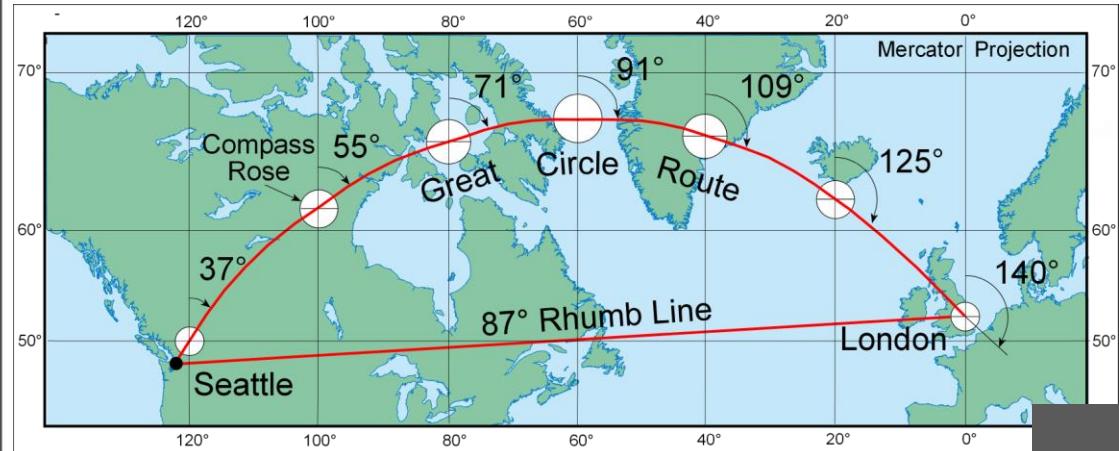
Models as abstractions rather than a 100% accurate representation

Mercator
projection



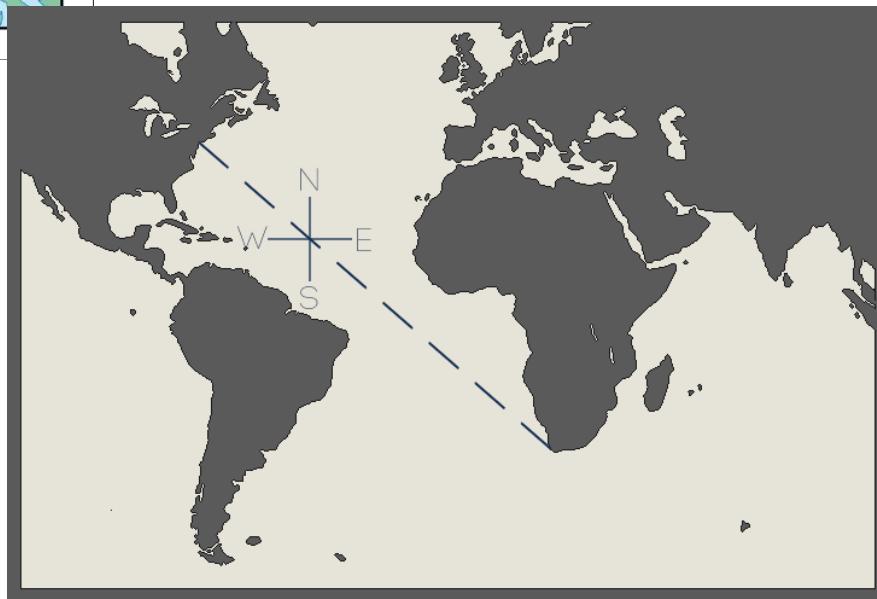


Models as abstractions rather than a 100% accurate representation



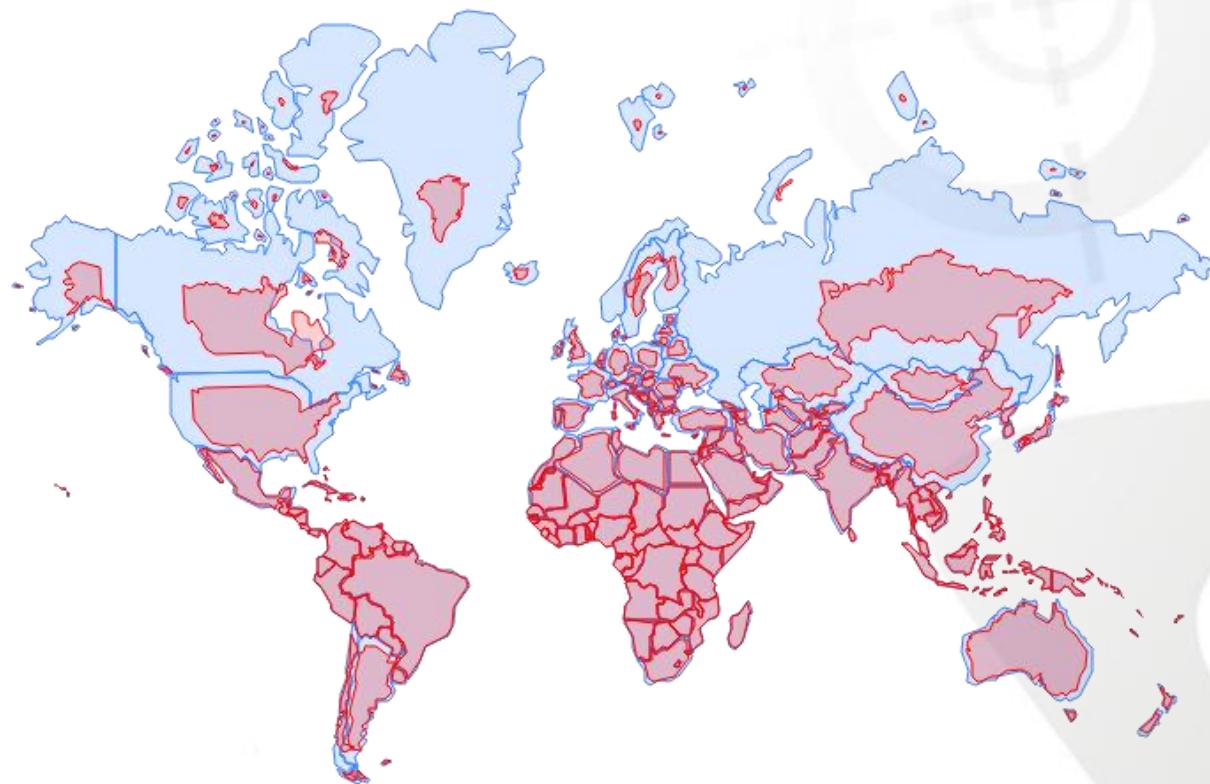
<https://medium.com/@pascal.sommer.ch/the-mercator-projection-doesnt-deserve-all-that-hate-d38537824f91>

https://en.wikipedia.org/wiki/Mercator_projection





Models as abstractions rather than a 100% accurate representation





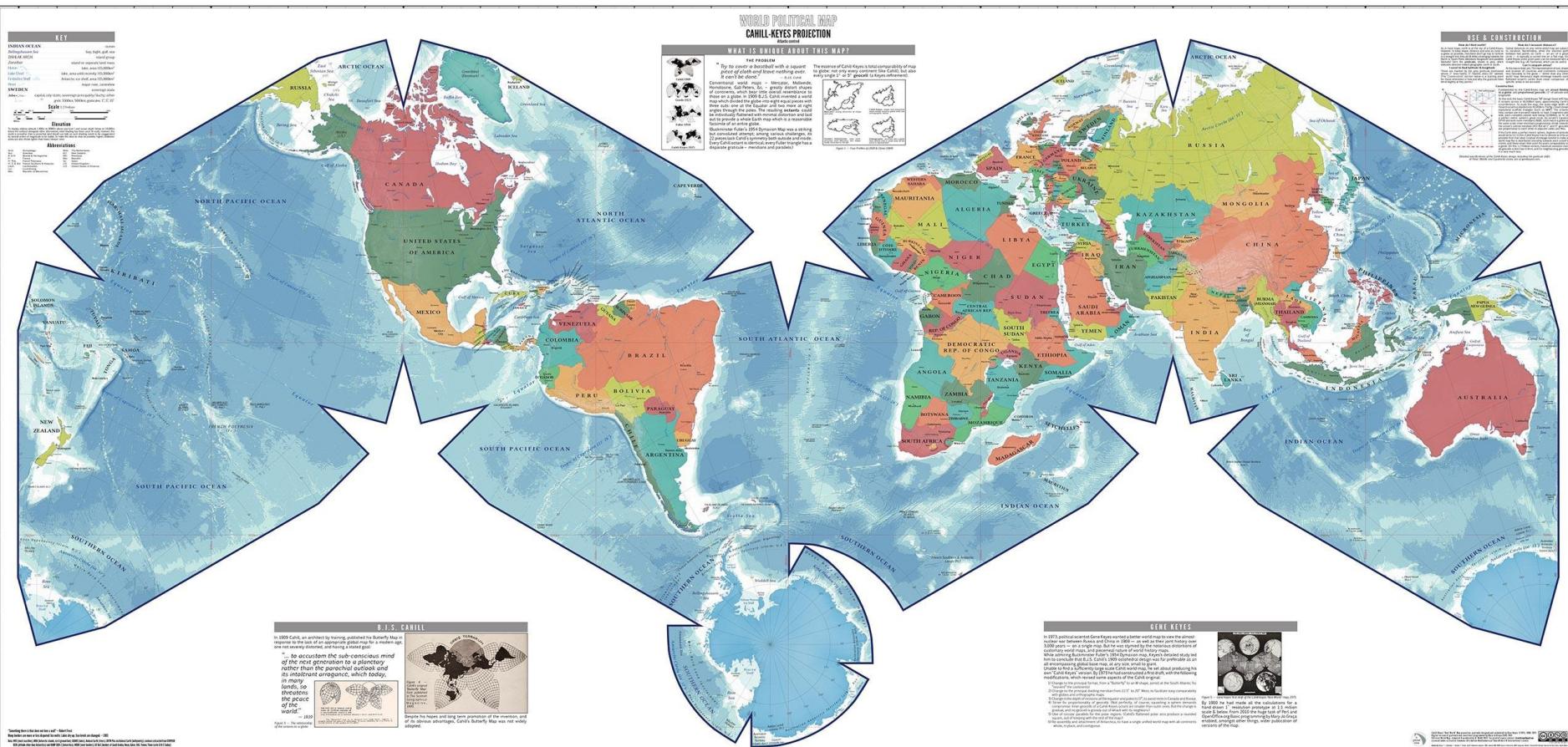
Models as abstractions rather than a 100% accurate representation



<https://brilliantmaps.com/xkcd/>



Models as abstractions rather than a 100% accurate representation



<http://gossipcity.org/actual-size-world-map/world-map-actual-size-new-the-true-size-africa-at-accurate-world-with-map-besttabletfor-of-world-map-actual-size-within-actual-size-world-map/>



Domain-Driven Design

*Software has to model the domain!
Software which does not have its
roots planted deeply into the domain
will not react well to change over
time.*

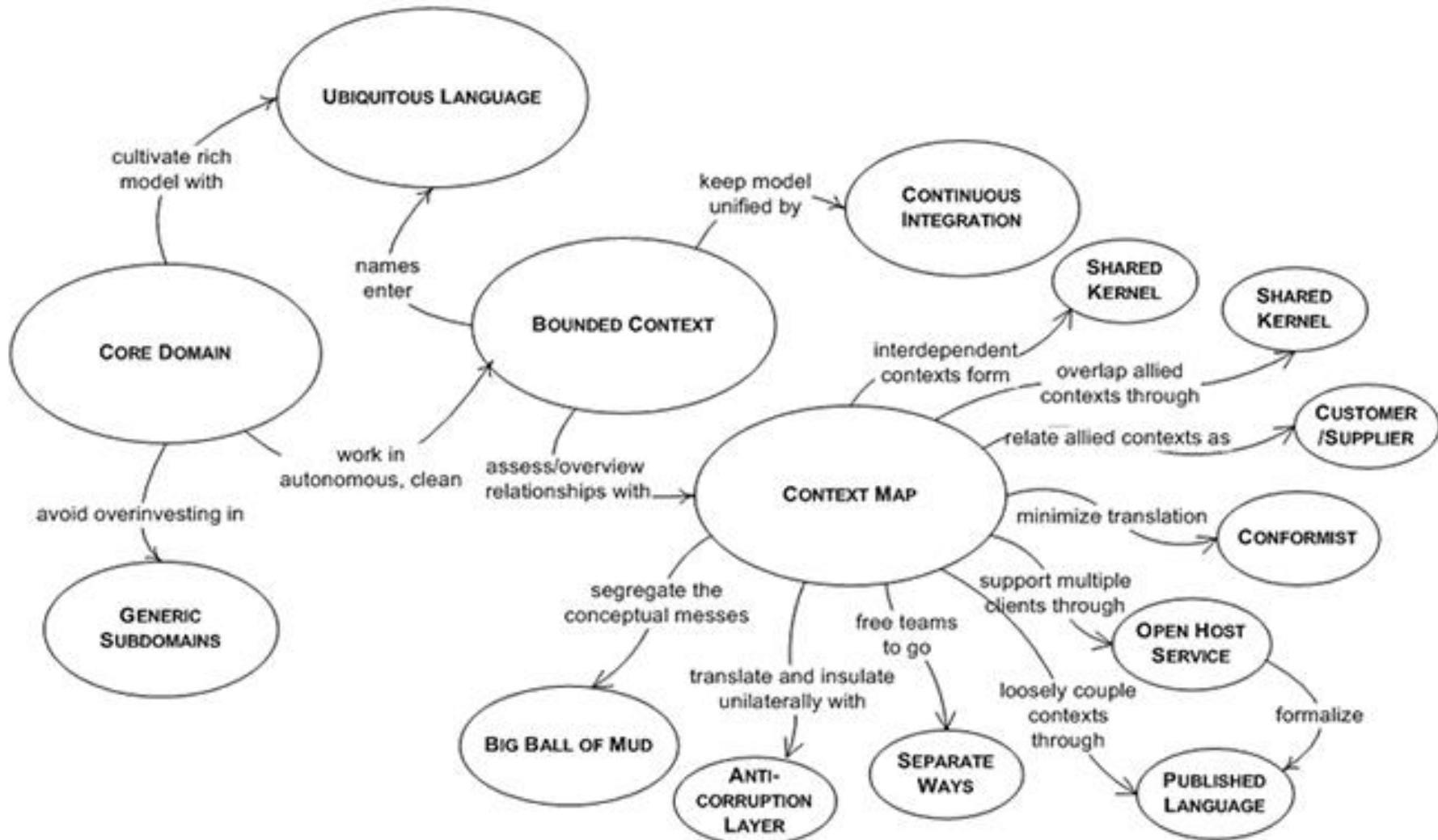


Domain-Driven Design



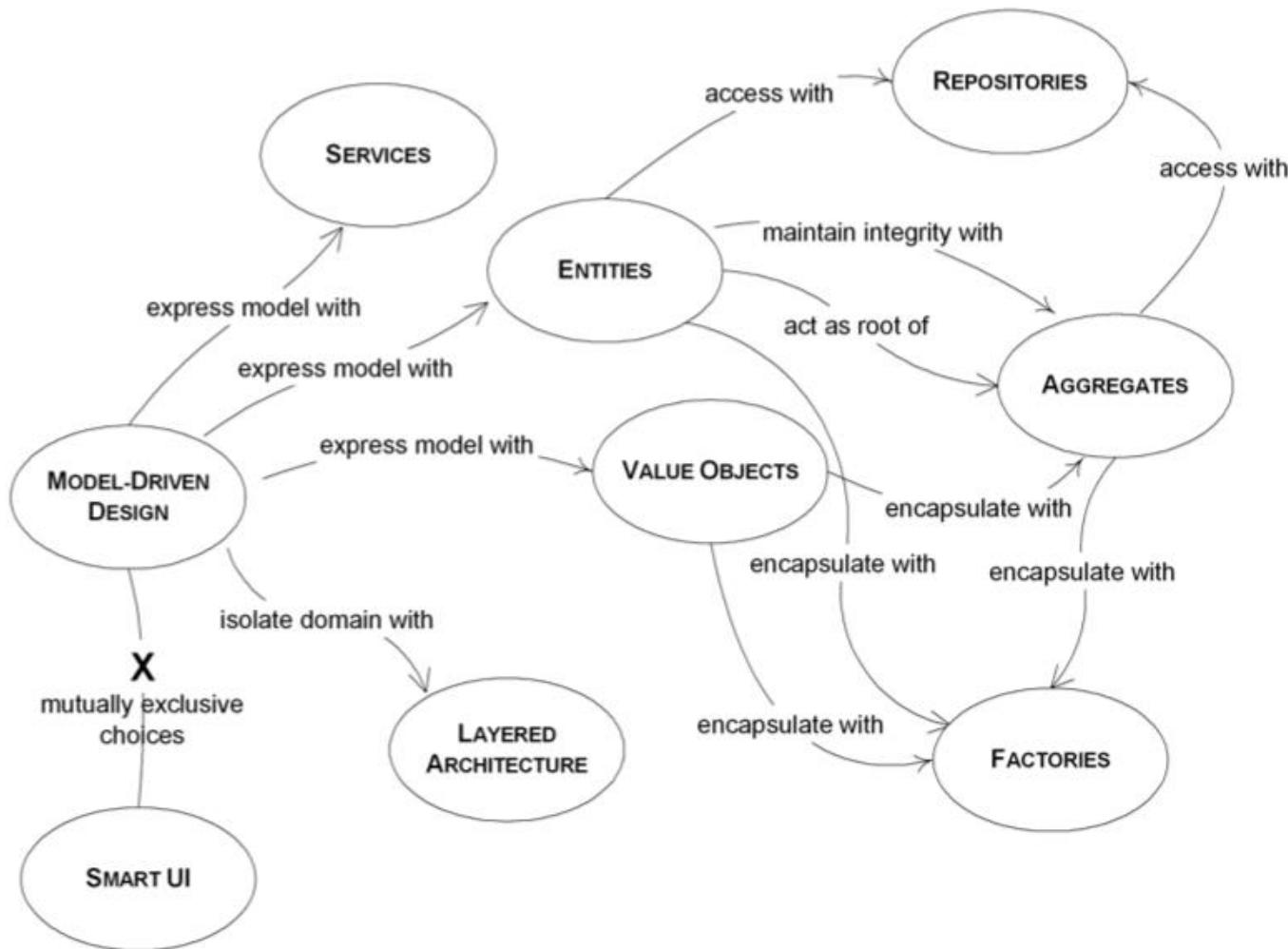


DDD – Strategic Design





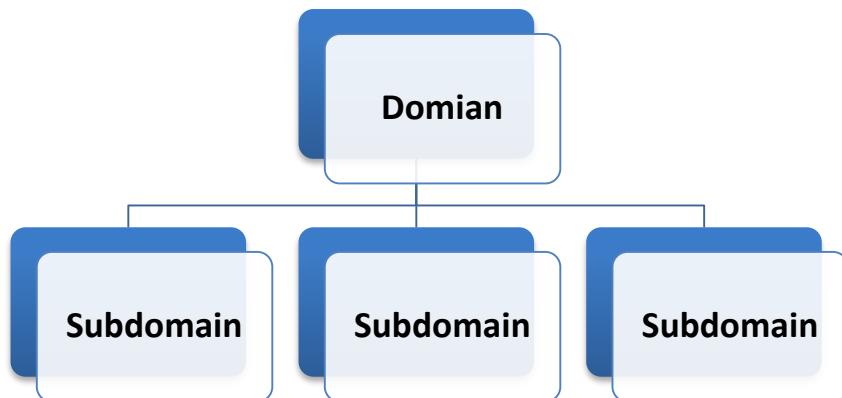
DDD – Tactical Design





DDD – Spaces

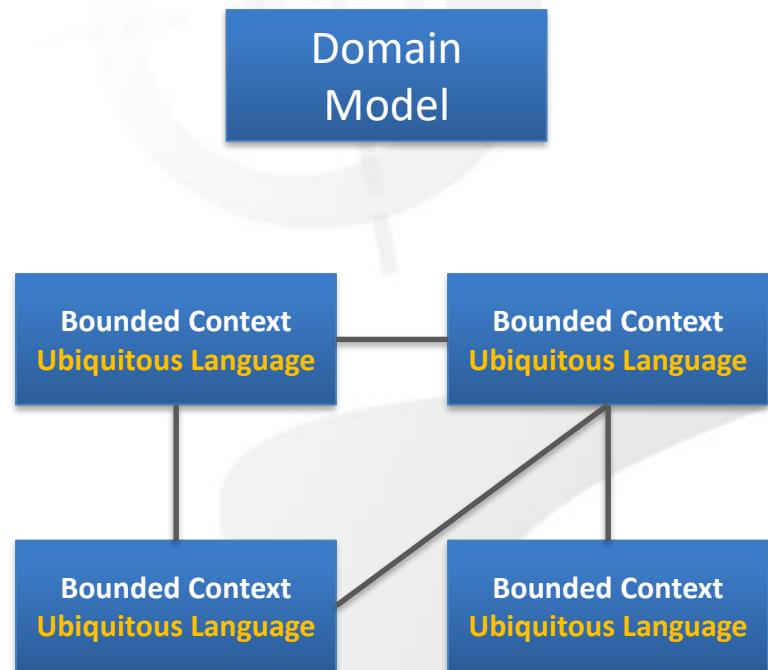
Problem Space



Tipos:

- Core Sub-Domain
- Supporting Sub-Domain
- Generic Sub-Domain

Solution Space





Domain-Driven Design – Event Storming





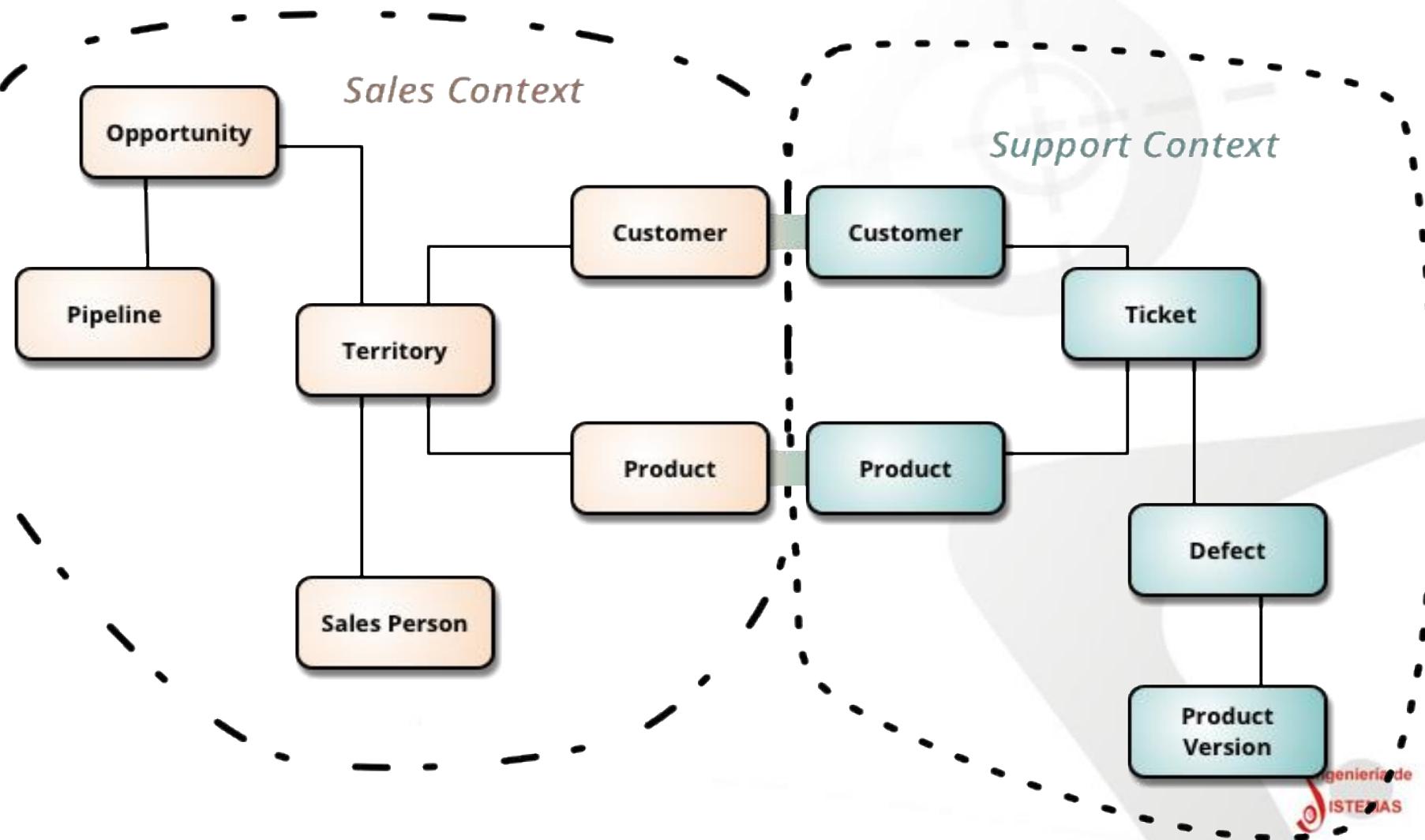
Domain-Driven Design – Event Storming

Subject -> Verb -> Object

- An item is added into the bill
- The user pays the bill



DDD – Bounded Context





DDD – Domain Activities

A bounded context API is comprised of Commands, Events and Queries.

- Commands (Requests)
- Events (Facts in the past)
- Queries (Like commands but without state change)



DDD – Domain Activities – Commands

Commands represent a request to change the state of the domain. They can be rejected and are usually phrased as a request.

A best practice is to define Commands that are intent revealing. This means that you can see from looking at the Command exactly what it was trying to do.

We should, therefore, favour Commands that are more specific, rather than more general.



DDD – Domain Activities – Commands

The user adds a new item to
his/her order



DDD – Domain Activities – Events

Events are often the result of a Command.

Where a Command requests a change to the state of the domain and Event records that change.

They often exist in a 1 to 1 relationship with the Command. Because they record something that happened in the past, they are best written as past tense.

Like Commands, Events are best when they reveal intent.



DDD – Domain Activities – Events

A new item was added to the user
order



DDD – Domain Activities – Queries

Queries can be done to obtain information about the state of the domain.

A query is issued in the form of a request for information, but it never changes the state of the domain.

If you perform the same query twice, without any other actions, then you should always get the same response.

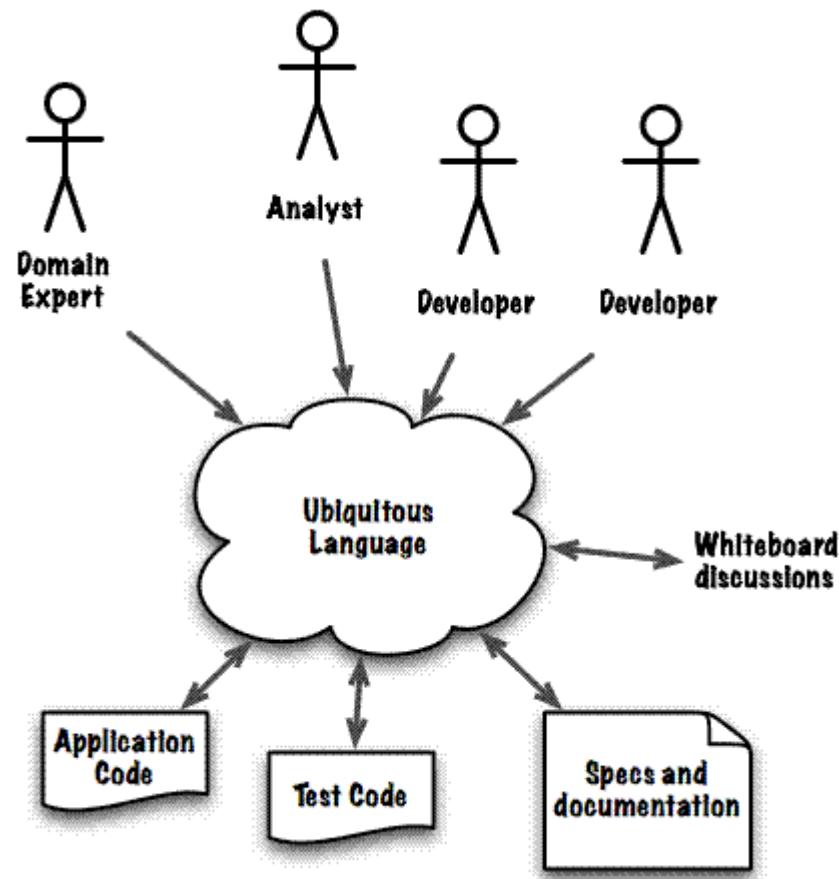


DDD – Domain Activities – Queries

The user sees all the items in his order



DDD – Ubiquitous Language





DDD – Domain Objects

- Entities (Unique Identity, mutable)
- Value Objects (Immutable, Usually used as a message between contexts)
- Aggregates:
 - Collection of domain objects
 - They have aggregate roots which are the point of access to the objects
 - A transaction shouldn't span more than one aggregate root
 - Objects in aggregate are treated as a unit



DDD – Domain Objects – Entities

User
id: 12345
firstName: German
lastName: Suarez

User
id: 67890
firstName: Alonso
lastName: Guerrero

User
id: 12345
firstName: Alonso
lastName: Suarez



DDD – Domain Objects – Value Objects

Phone

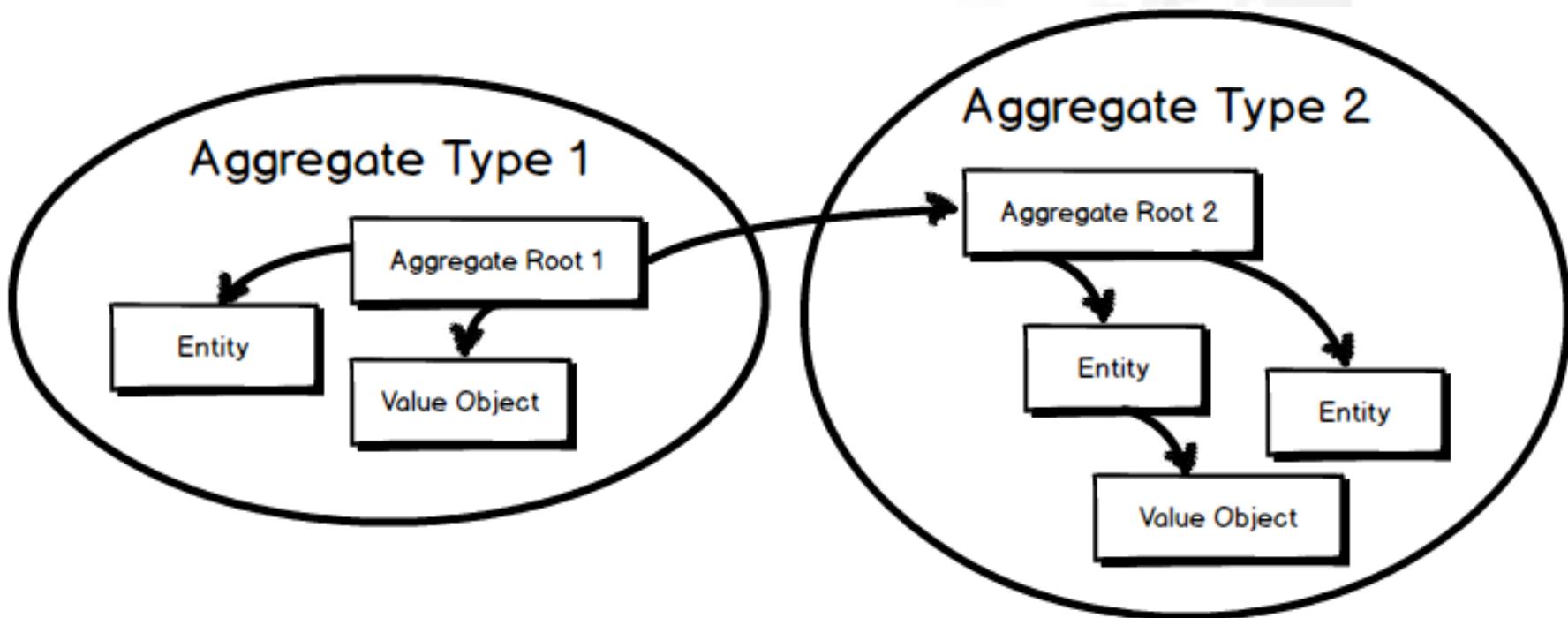
number: 1231268

areaCode: 571

countryISOCode: CO

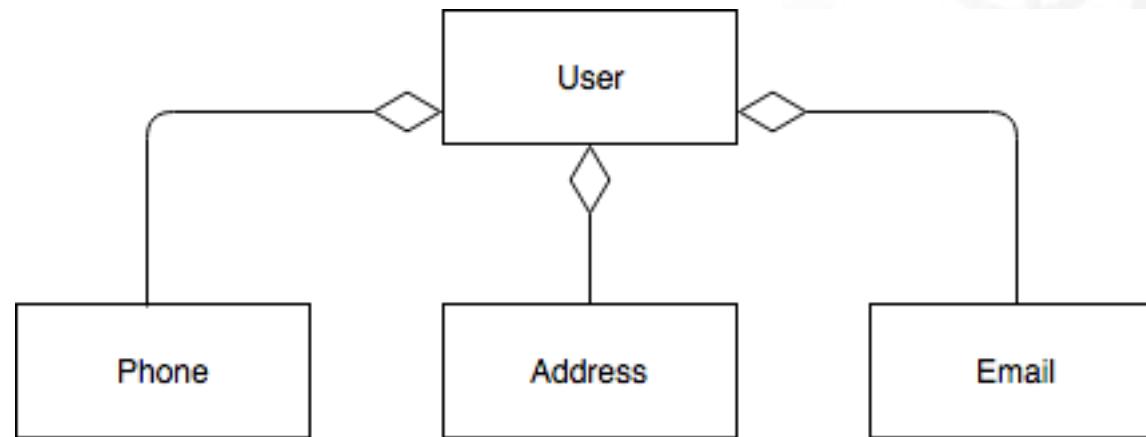


DDD – Domain Objects – Aggregates





DDD – Domain Objects – Aggregates





DDD – Stringify Issue

```
public class User {  
  
    private int id;  
    private String firstName;  
    private String lastName;  
    private List<Long> phones;  
    private String email;  
}
```



DDD – Stringify Issue

```
final public class FirstName {  
  
    @NotNull  
    @NotEmpty  
    @Pattern(regexp = "[a-zA-Z]+")  
    private String value;  
  
    public FirstName(@NotNull @NotEmpty @Pattern(regexp  
= "[a-zA-Z]+") String value) {  
        this.value = value;  
    }  
  
    public String getValue() {  
        return value;  
    }  
}
```



DDD – Stringify Issue

```
final public class Phone {  
  
    @NotNull  
    private Long number;  
    @NotNull  
    private Long areaCode;  
    @NotNull  
    @Pattern(regexp = "[a-zA-Z]{2} | [0-9]{3}")  
    private String countryISOCode;  
    @NotNull  
    private boolean isCellphone;  
  
    public Phone(@NotNull Long number, @NotNull Long areaCode,  
    @NotNull @Pattern(regexp = "[a-zA-Z]{2} | [0-9]{3}") String  
    countryISOCode, @NotNull boolean isCellphone) {  
        this.number = number;  
        this.areaCode = areaCode;  
        this.countryISOCode = countryISOCode;  
        this.isCellphone = isCellphone;  
    }  
}
```



DDD – Stringify Issue

```
final public class Email {  
  
    @NotNull  
    @Pattern(regexp = "(?![a-zA-Z0-  
9!#$%&'*/=...])")  
    private String value;  
  
    public Email(@NotNull @Pattern(regexp =  
"(?![a-zA-Z0-9!#$%&'*/=...])" String value) {  
        this.value = value;  
    }  
}
```



DDD – Stringify Issue

```
public class User {  
  
    private String firstName;  
    private String lastName;  
    private List<Long> phones;  
    private String email;  
}
```



DDD – Stringify Issue

```
public class User {  
  
    private FirstName firstName;  
    private LastName lastName;  
    private List<Phone> phones;  
    private Email personalEmail;  
}
```



DDD – Domain Abstractions

- Services
- Factories
- Repositories



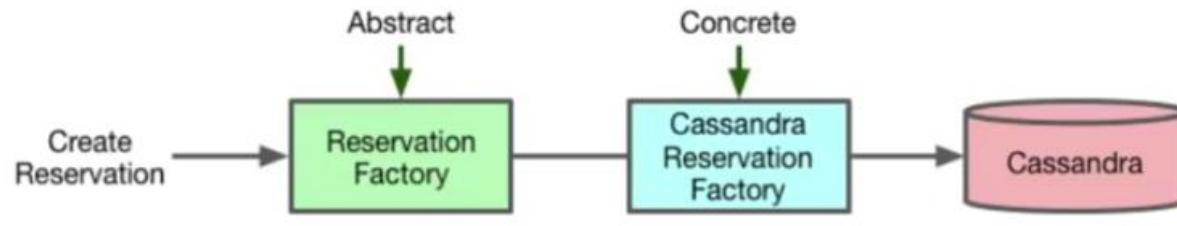
DDD – Domain Abstractions – Services

- Business logic that doesn't fit with an Entity or Value Object
- Should be stateless
- Often used to abstract away anti-corruption layer
- Anemic domains should be avoided



DDD – Domain Abstractions – Factories

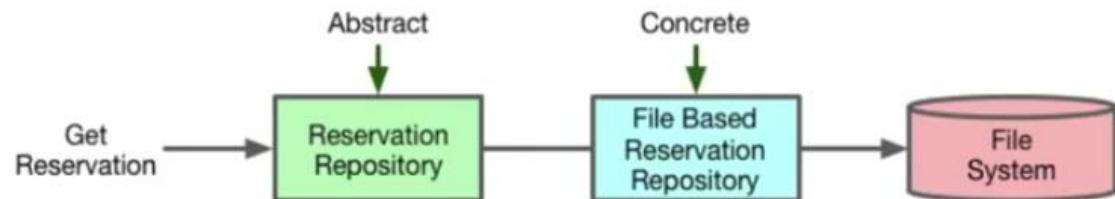
- The logic to construct new domain objects may not be trivial
- May require access to external resources (DBs, files, REST APIs, etc.)
- Factories abstract away the logic of creation
- Usually implemented as a domain interfaces, with one or more concrete implementations
- They are the **C** in **CRUD**





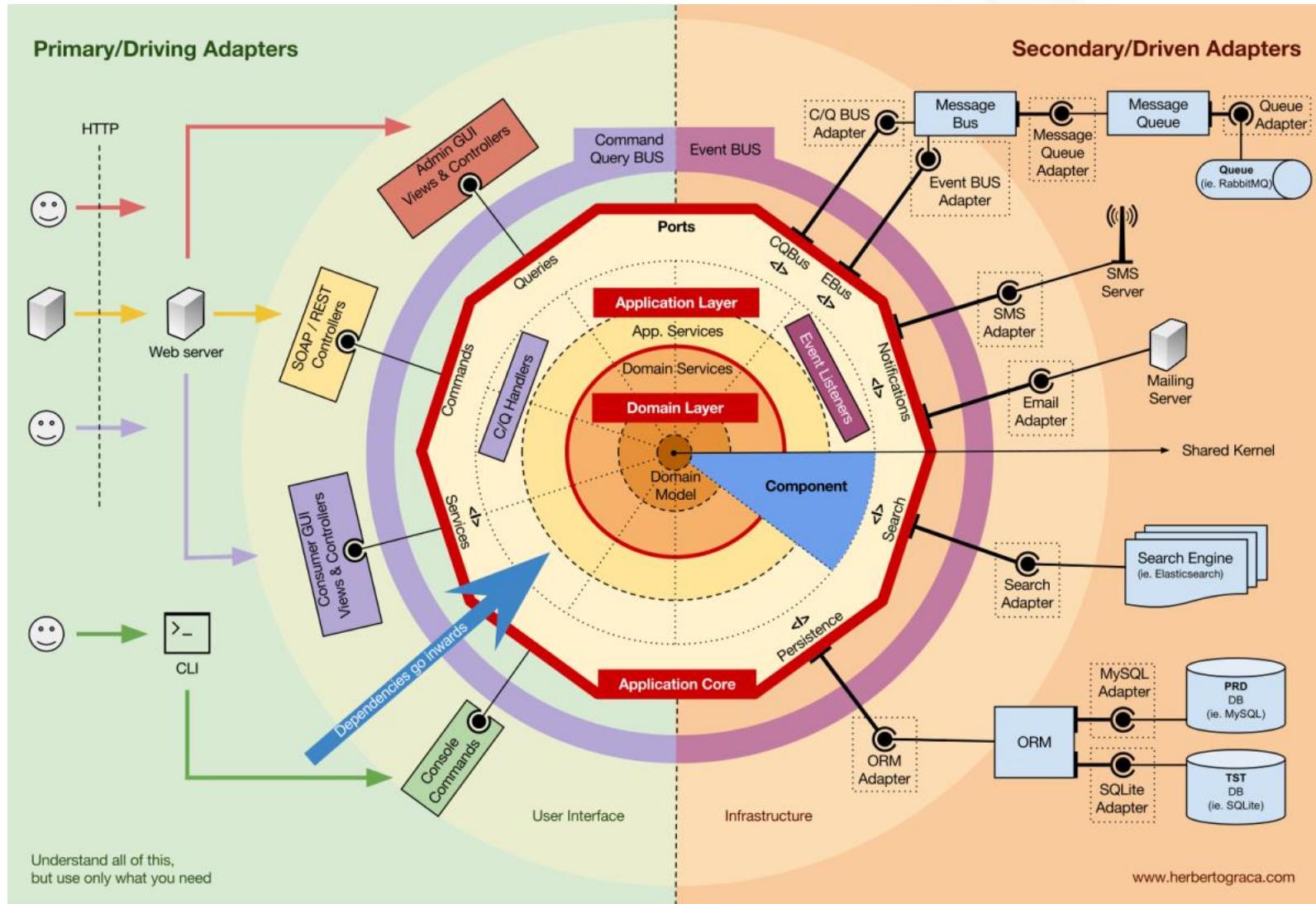
DDD – Domain Abstractions – Repositories

- Similar to Factories, Repositories abstract away the retrieving of existing objects
- Factories are used to get new objects, Repositories are used to get, or modify existing objects
- Often operate as abstractions layers over DBs, but they can work with files, REST APIs, etc.
- A Repository doesn't imply a DB
- They are the **RUD** in **CRUD**





DDD - Hexagonal Architecture





Referencias

- Rittgen, Peter: “Enterprise Modeling and Computing with UML”. IGI Publishing, 2008.
- “Diccionario de la Real Academia Española”. IGI Publishing, ESPASA, 22^a edición, 2001.
- <http://dddcommunity.org>
- <https://www.infoq.com/articles/ddd-contextmapping>
- <https://martinfowler.com/tags/domain%20driven%20design.html>
- <https://www.martinfowler.com/bliki/AnemicDomainModel.html>
- <https://vladikk.com/2016/04/05/tackling-complexity-ddd/>