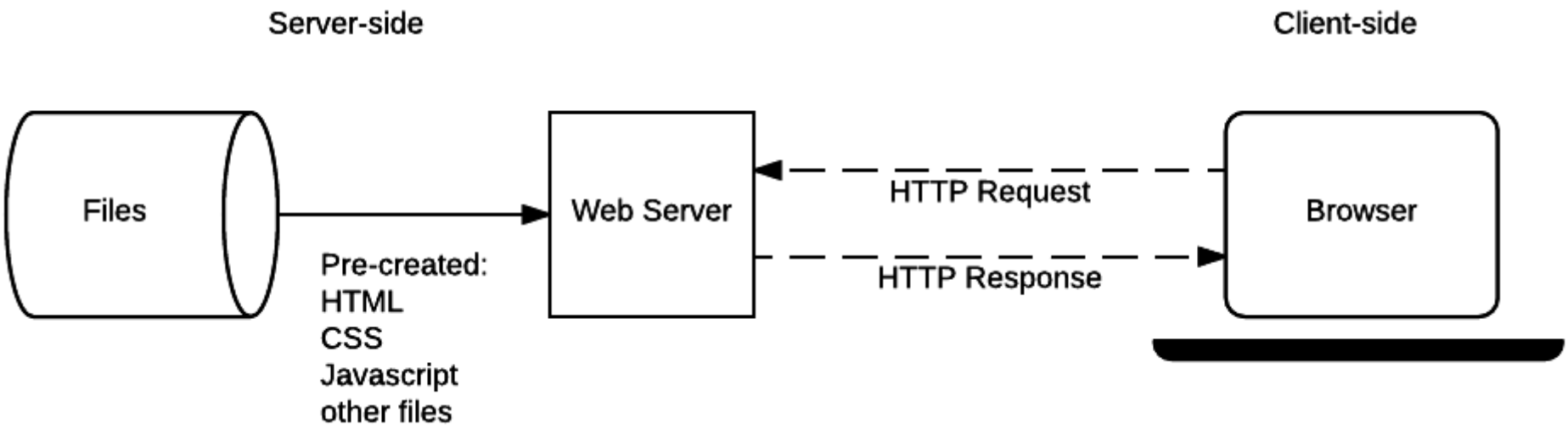
MICROSERVICES

Stefano Bruna

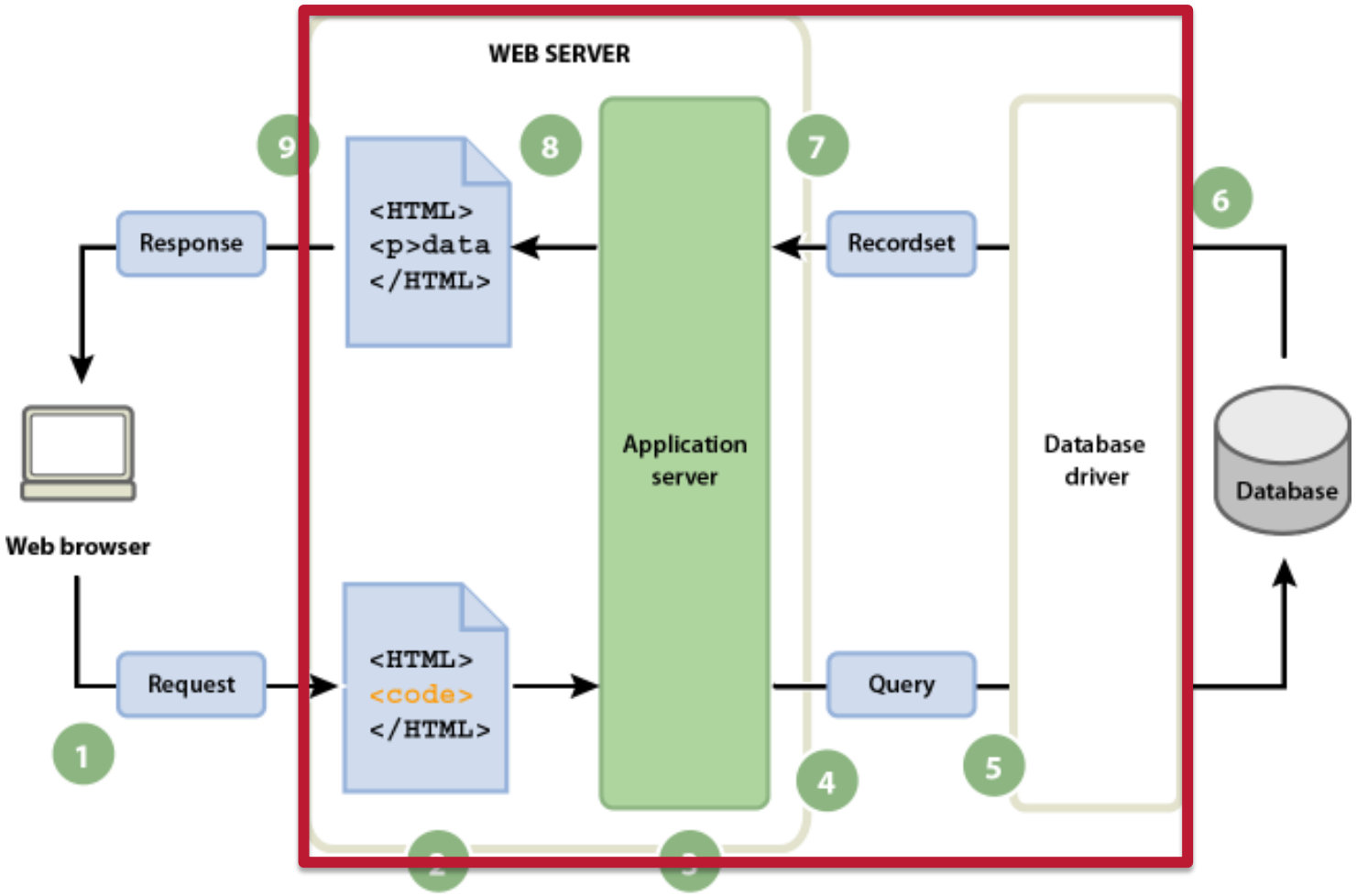
Agenda

- Introduction about web application and architecture
- The monolith architectural approach
- Technical debt
- The microservice paradigm

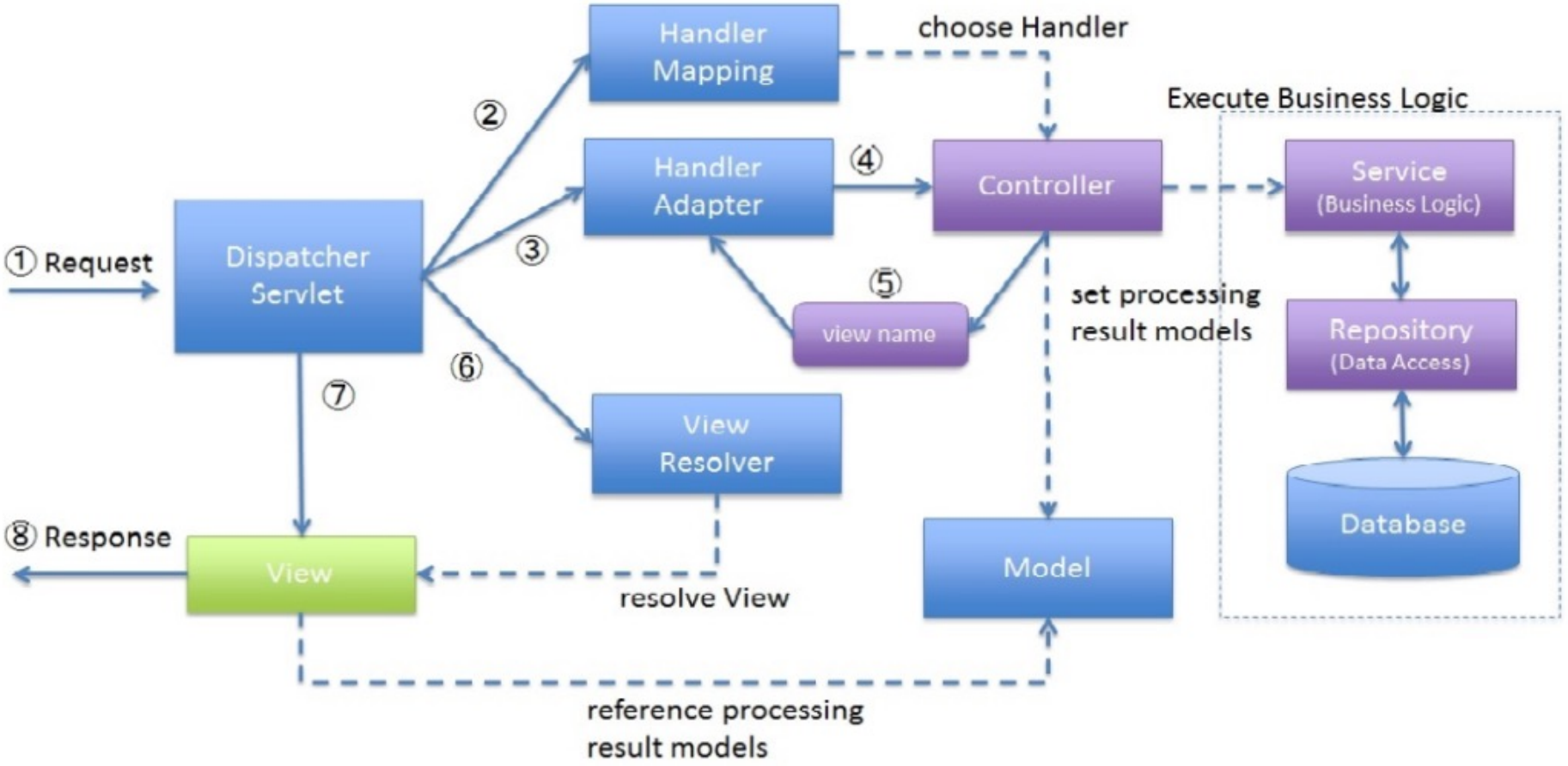
Static Web Sites



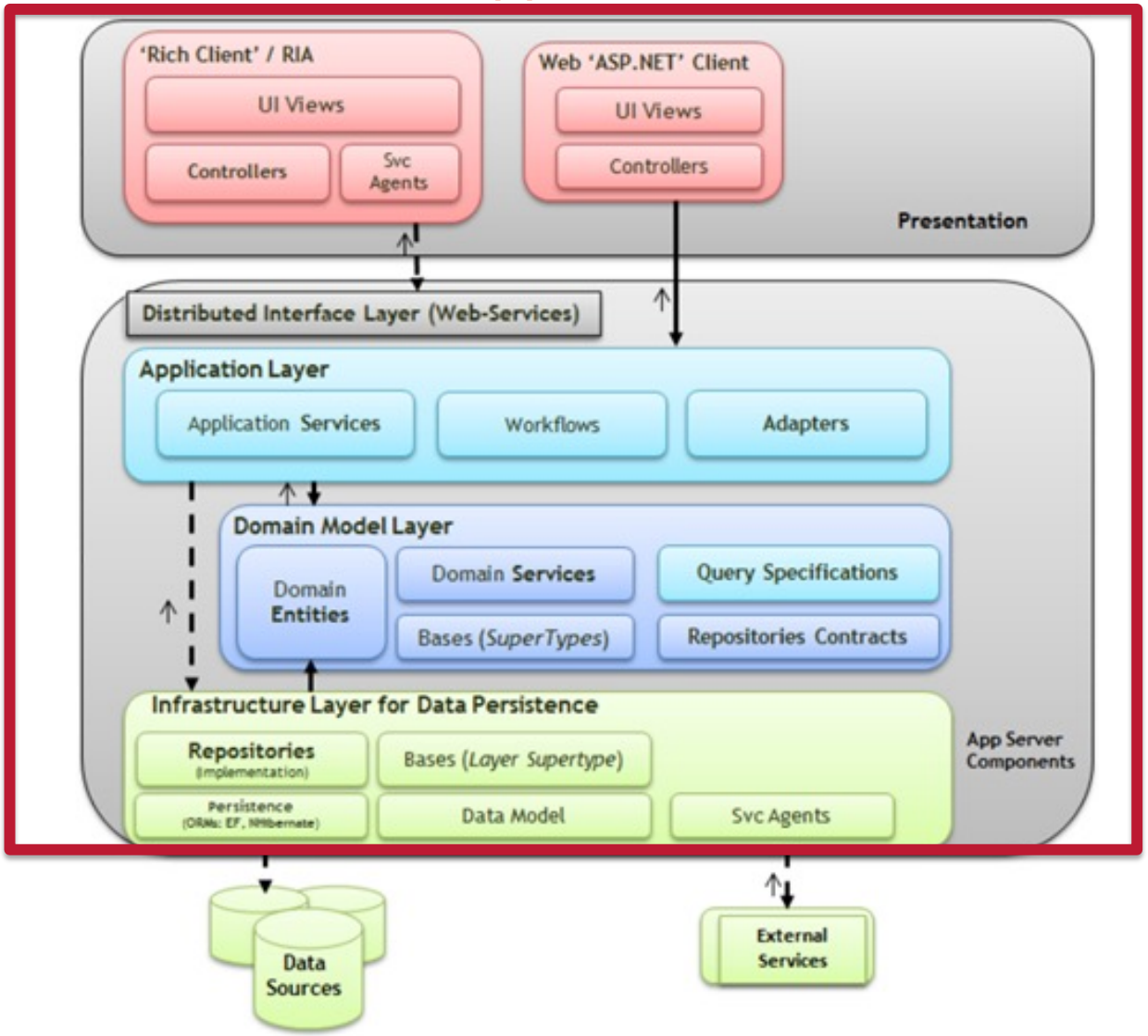
Web Applications



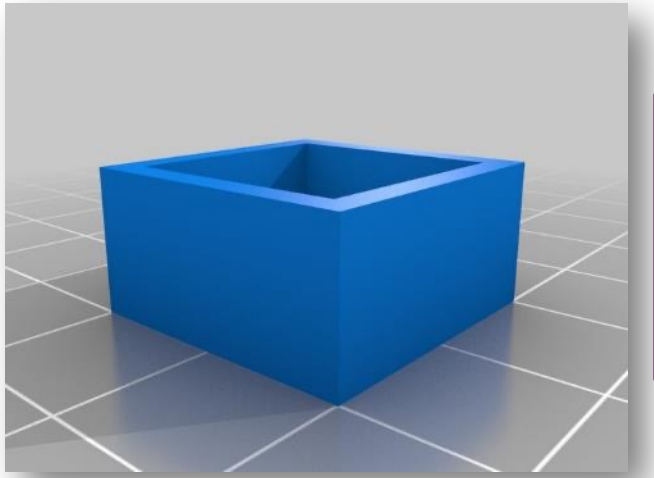
How to manage complexity ?
MVC design pattern



DDD web application architecture



Monolithic architectural paradigm

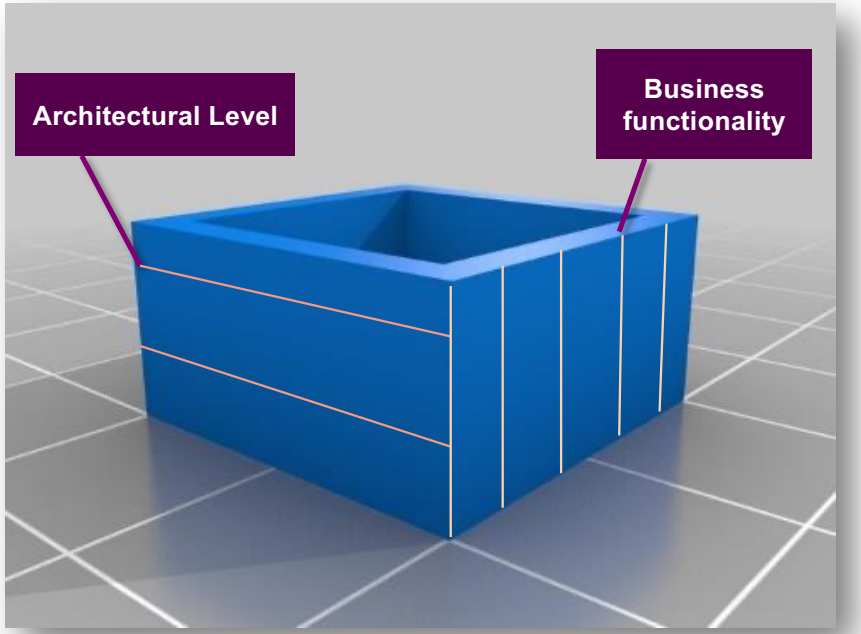


A single executable macro-component that contains all the application logic

POLITECNICO MILANO 1863

webscience an adesso company

Monolithic architectural paradigm



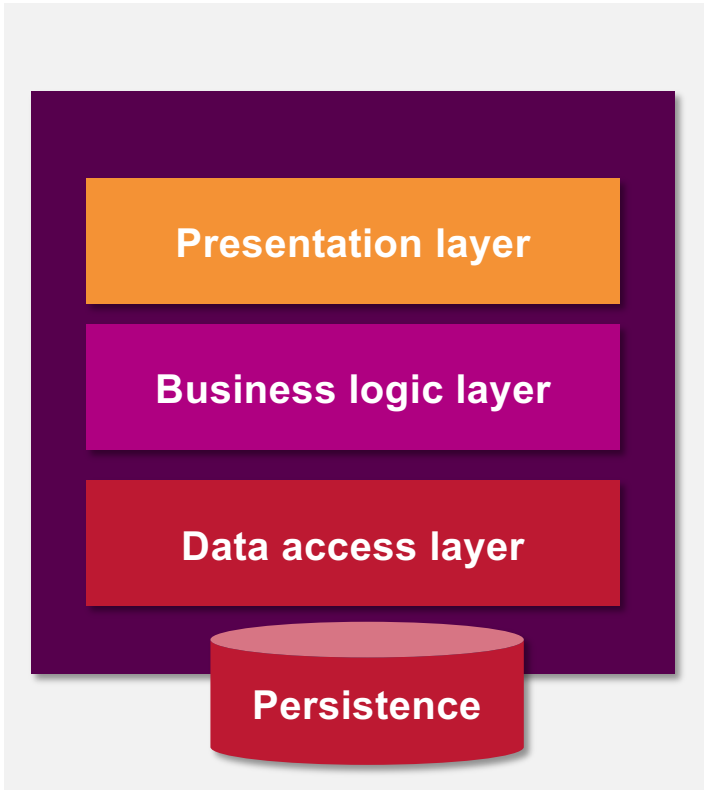
Tipically composed of modules that try to encapsulate the application logics.

Two orthogonal dimensions:

- Architectural level
- Business functionality

Architectural level

The **architectural level** dimension usually consists of three levels (layers)



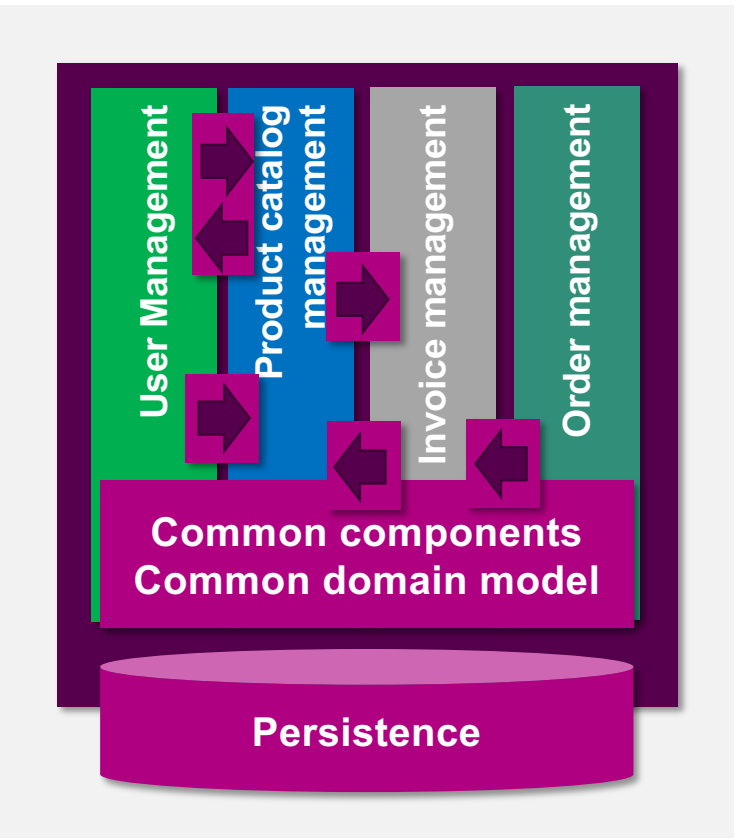
Business functionalities

The size of the **application functionality** includes, for example:

- User Management
- Product catalog management
- Invoice management
- Order management

...

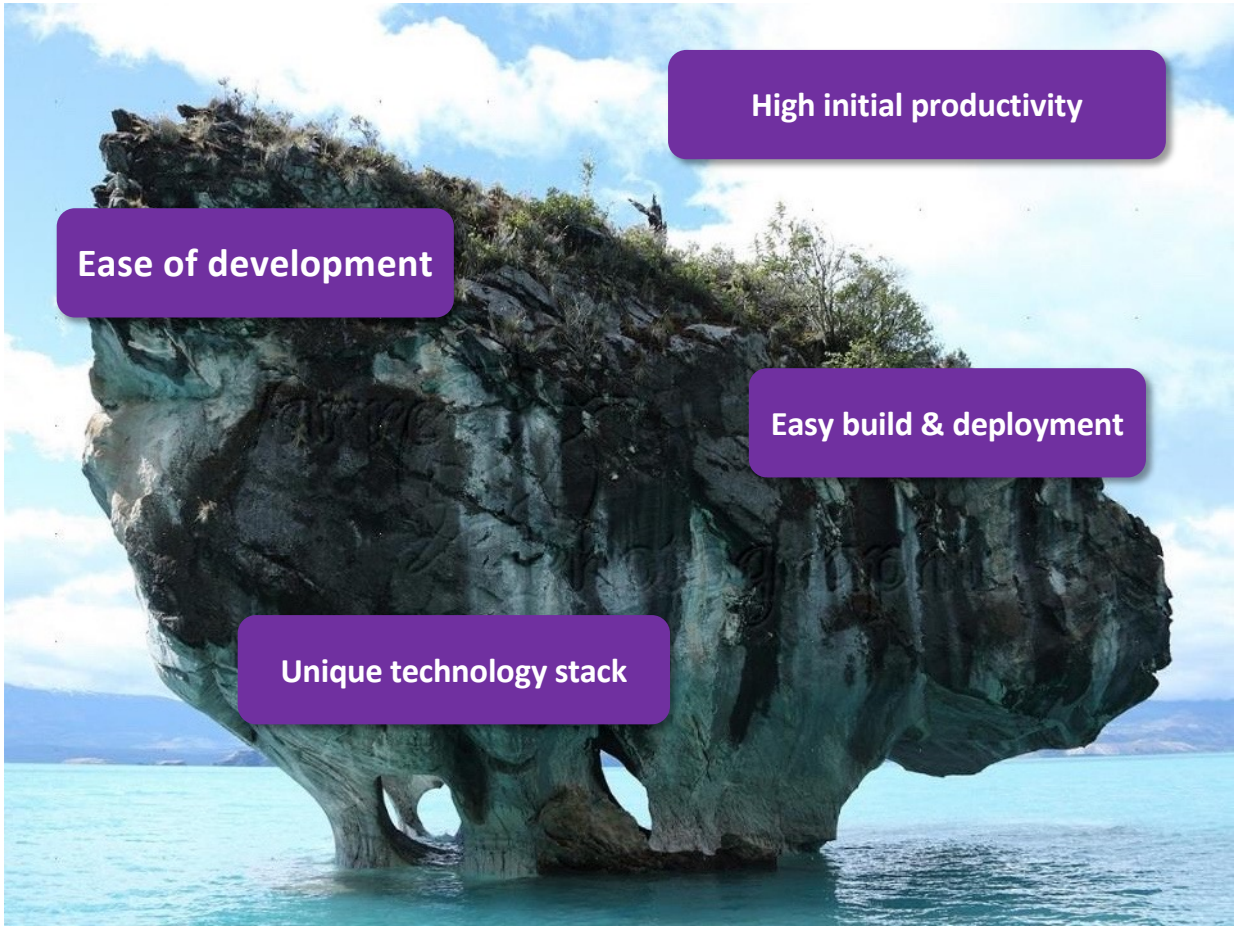
The functionalities are highly coupled to each other



POLITECNICO MILANO 1863

webscience an adesso company

The strengths of monolithic architecture

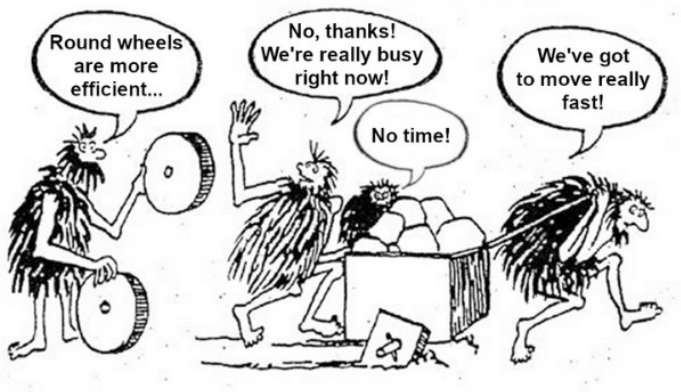


POLITECNICO MILANO 1863

webscience an adesso company

Source: https://inspirezone.tech/what-is-technical-debt/

What is technical debt ?

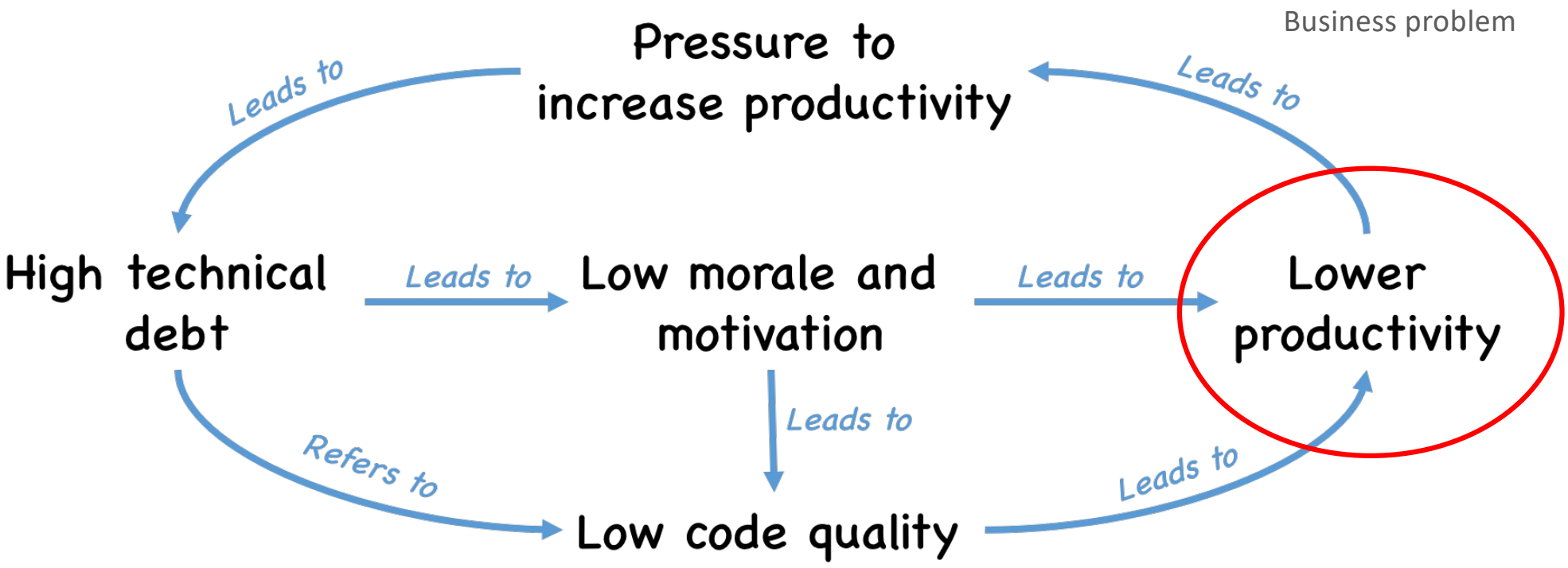


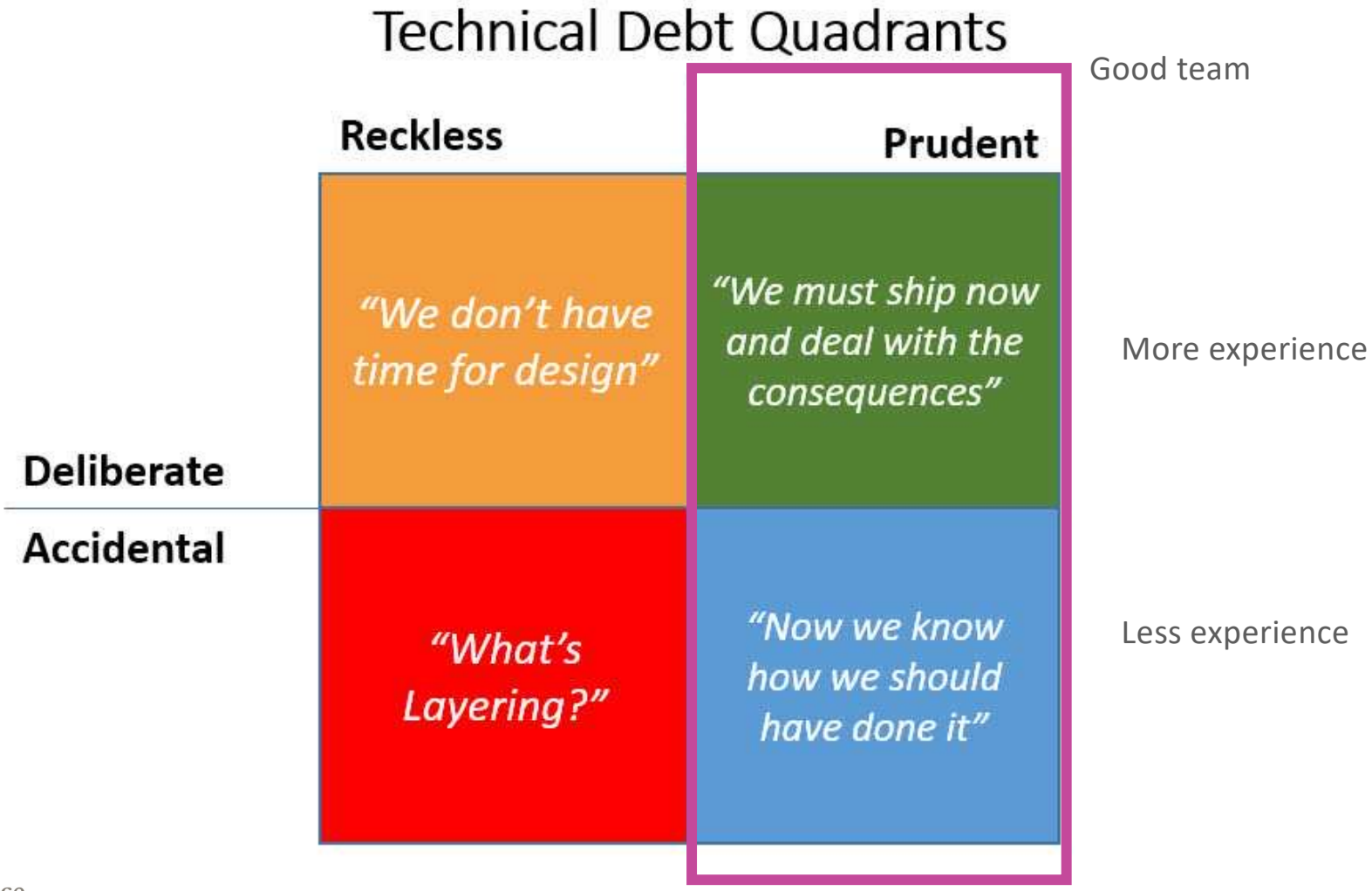
«Technical debt is a term used in software development processes describing trade-off between choosing a **quicker short-term solution** that results in long-term consequences. Like financial debt, technical debt accumulates interest in the form of the **extra effort to overcome problems** that appear in the future»

Different forms:

- **Requirements:** The gap between the captured product requirements and what was actually implemented.
- **Versioning:** Problems that arise with not using a clear versioning system or having a disorganised repository.
- **Build:** Problems with the building process that make it complex or difficult.
- **Test:** Any shortcuts taking in testing or a lack of tests.
- **Defect:** Any bugs or failures found in the software.
- **Design: Technical shortcuts taken during the design stage.**
- **Architectural:** Decisions made that compromise quality and maintainability.
- **Documentation:** Refers to documents on the project that are out-of-date, incomplete or a code base that lacks comments.
- **Code:** Refers to poorly written code that doesn't follow best coding practices.
- **Infrastructure:** Using tools, technologies or configuration that are not optimal for the software solution.

Technical debt dynamic

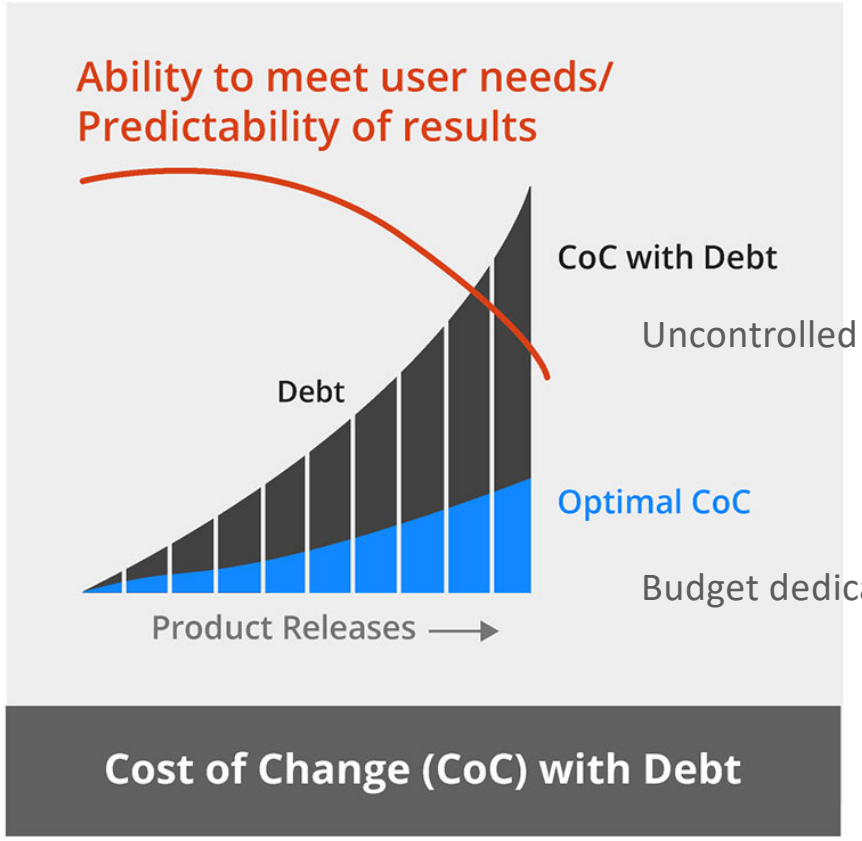




POLITECNICO MILANO 1863

webscience an adesso company

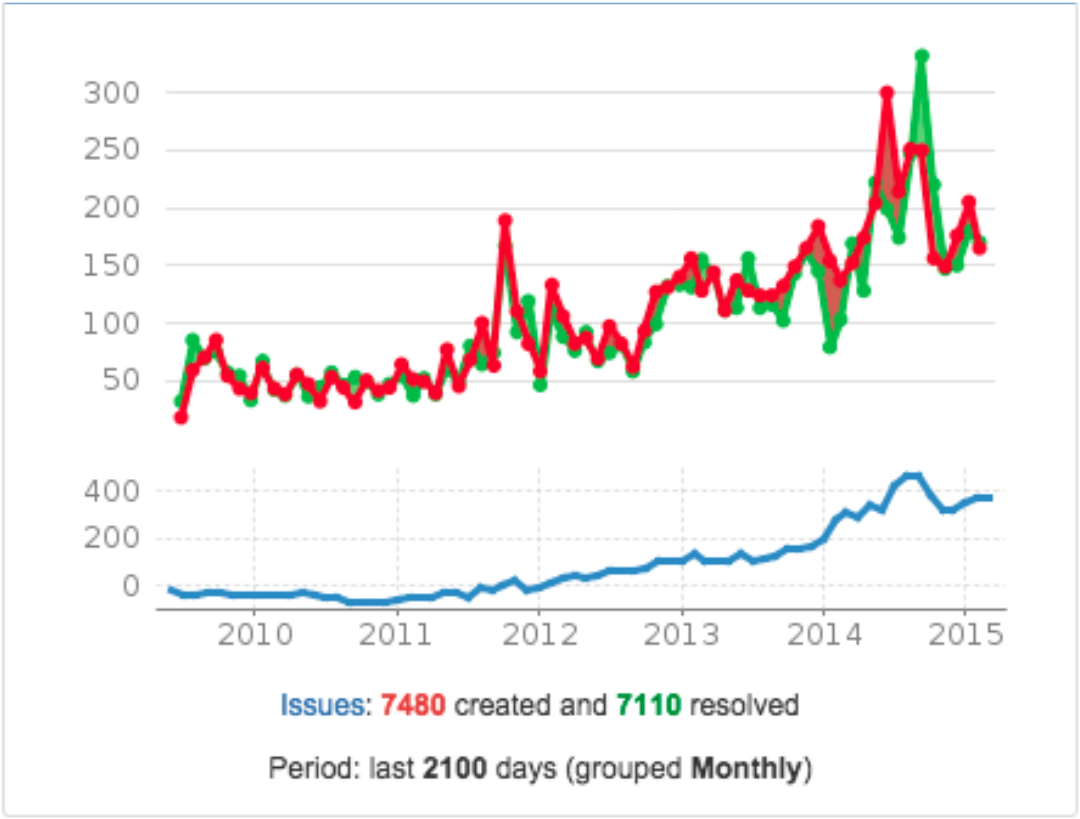
Technical debt in complex system



POLITECNICO MILANO 1863

webscience an adesso company

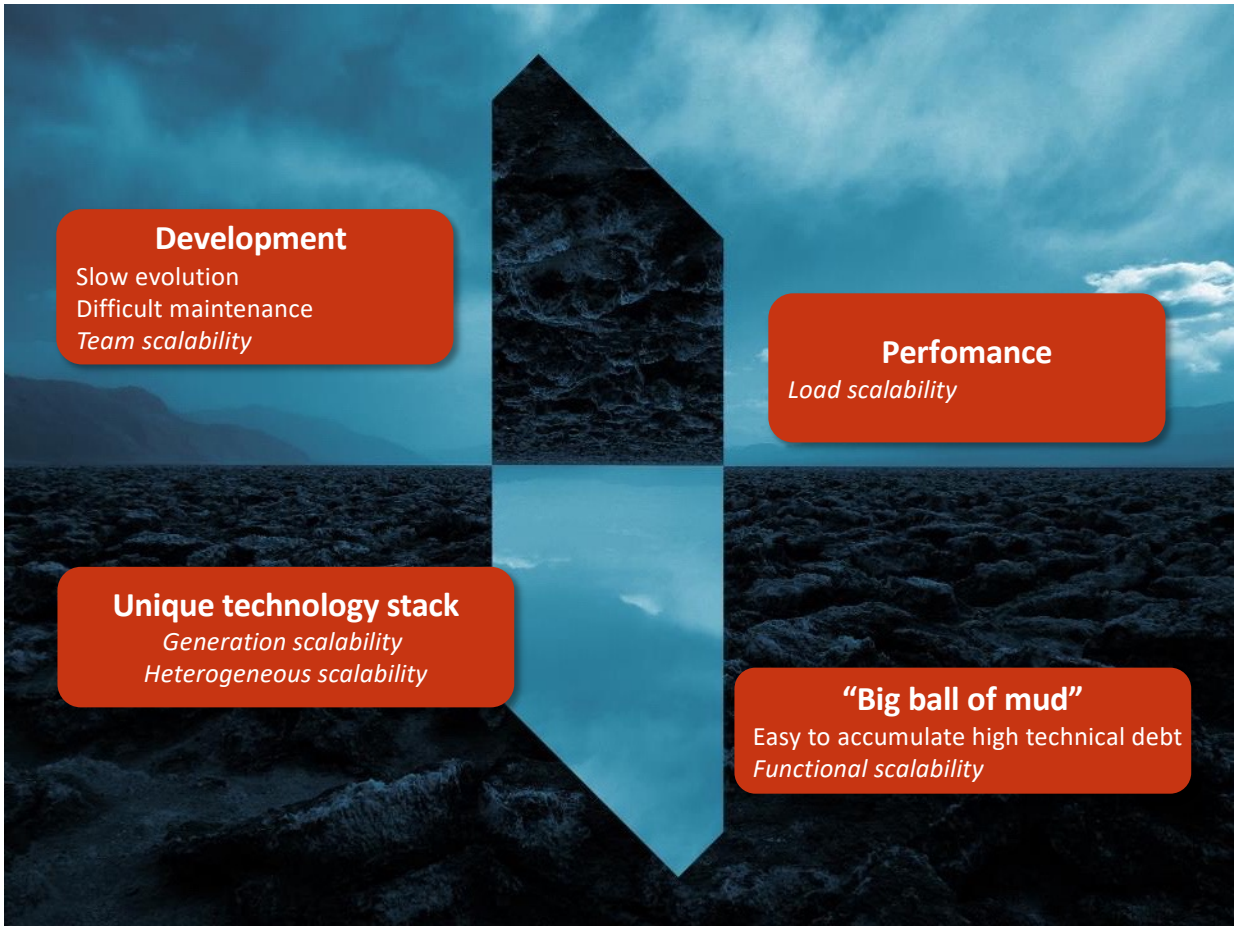
Application maintenance



POLITECNICO MILANO 1863

webscience an adesso company

The weaknesses of monolithic architecture



POLITECNICO MILANO 1863

webscience an adesso company

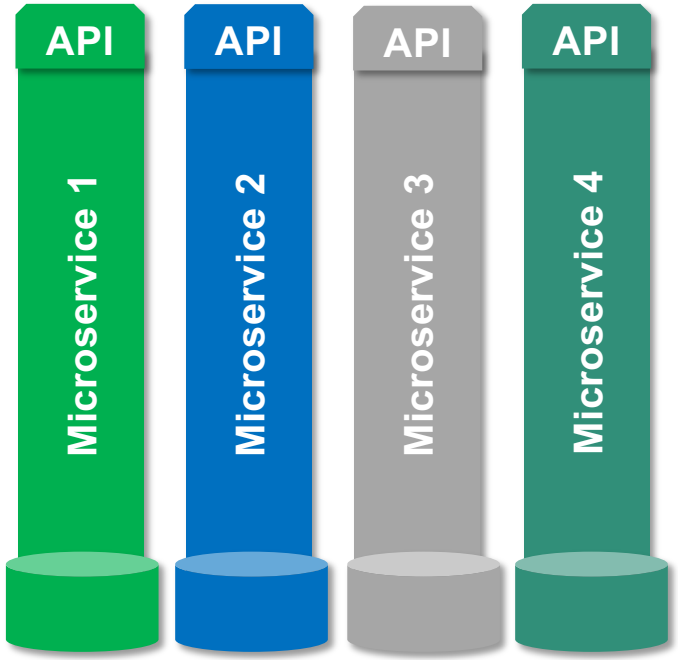
Microservices architecture, an alternative paradigm

POLITECNICO MILANO 1863

webscience an adesso company

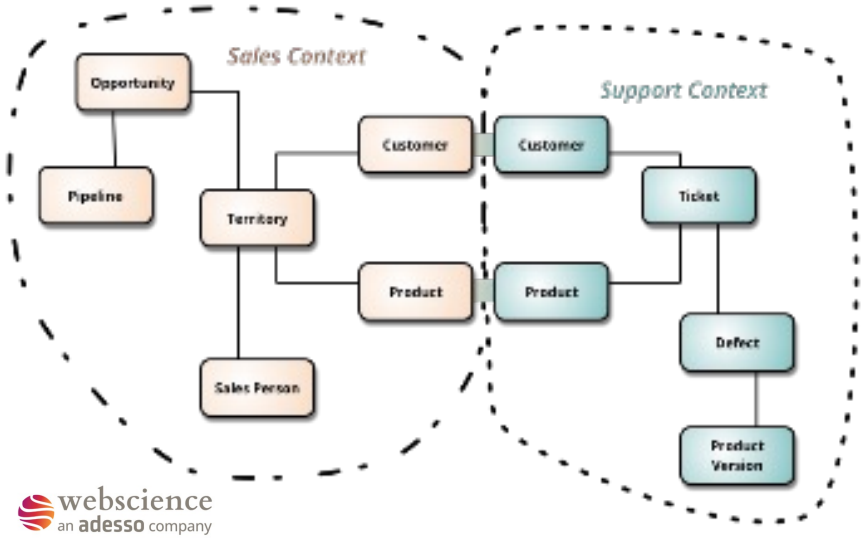
Logical architecture

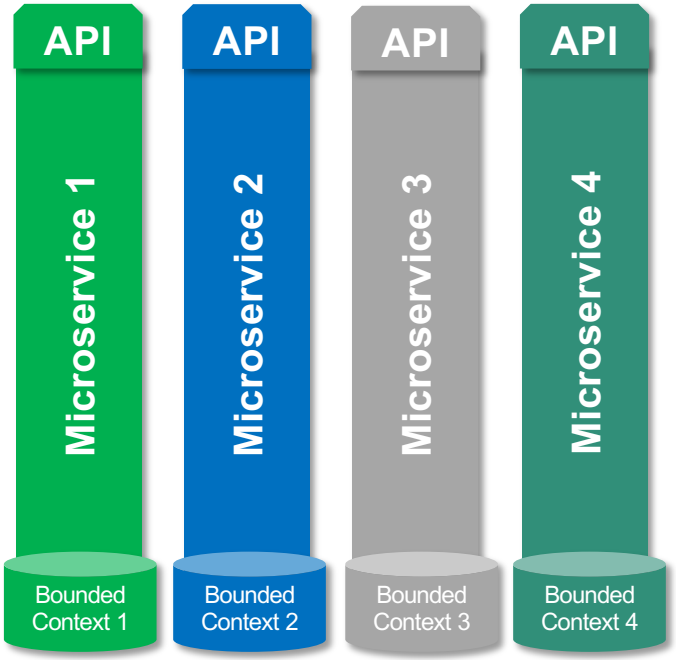
- **Application developed as:**
  - suite of small executable units (microservices)
  - who publish business oriented API
- Philosophy : **“Do one thing and do it well”**



Logical architecture: bounded context

- Each microservice manages a consistent subset of functionalities (processes and entities) belonging to the global application domain.
- Each microservice manages a "bounded context": a piece of the global domain that is sufficiently atomic and independent from the rest of the application domain



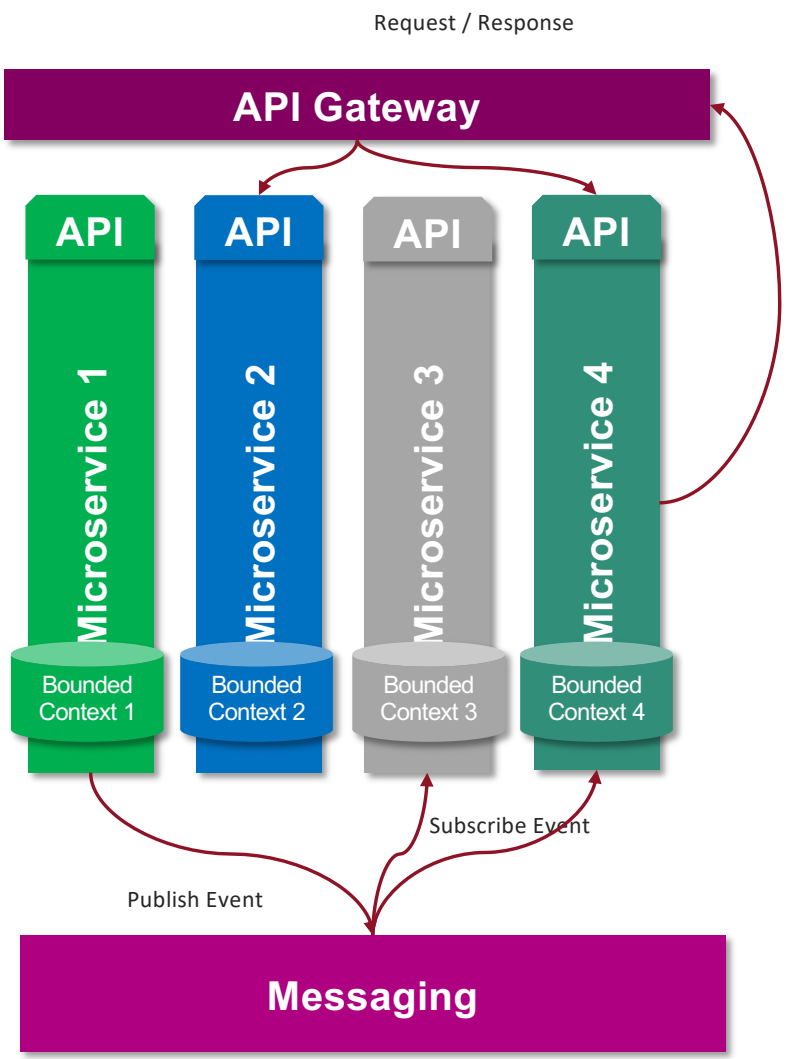


POLITECNICO MILANO 1863

webscience an adesso company

Logical architecture: collaboration

- Microservices are **loosely coupled** to each other
- Microservices **collaborate** with each other using "lightweight" protocols:
  - Request / Response Http
  - Publish / Subscribe Messaging

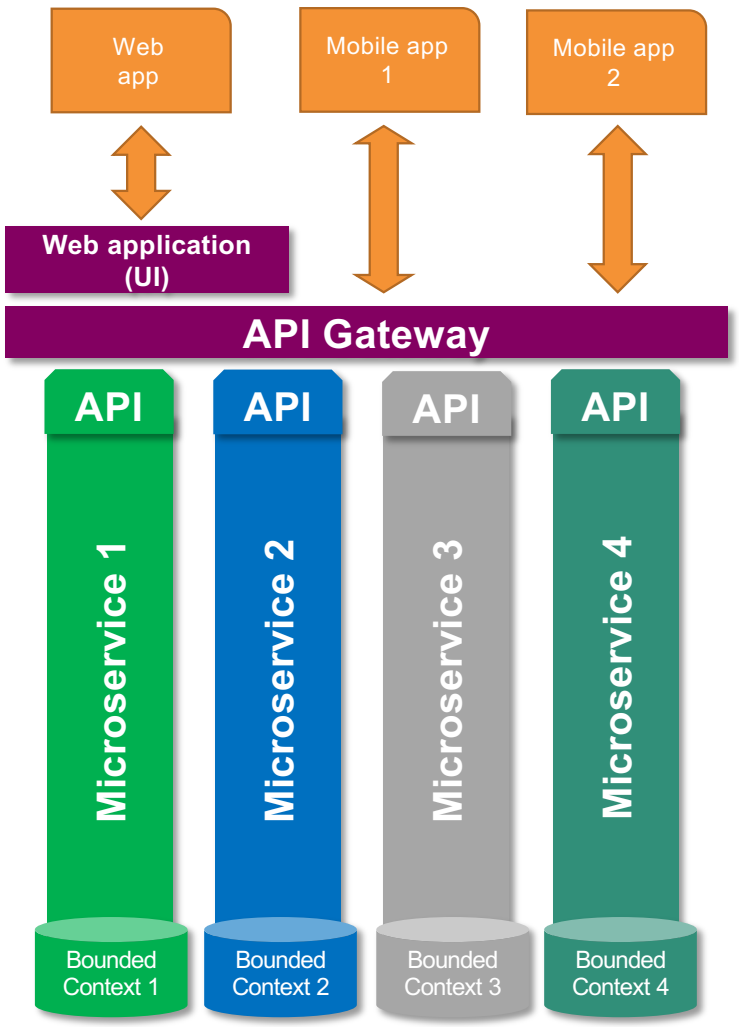


POLITECNICO MILANO 1863

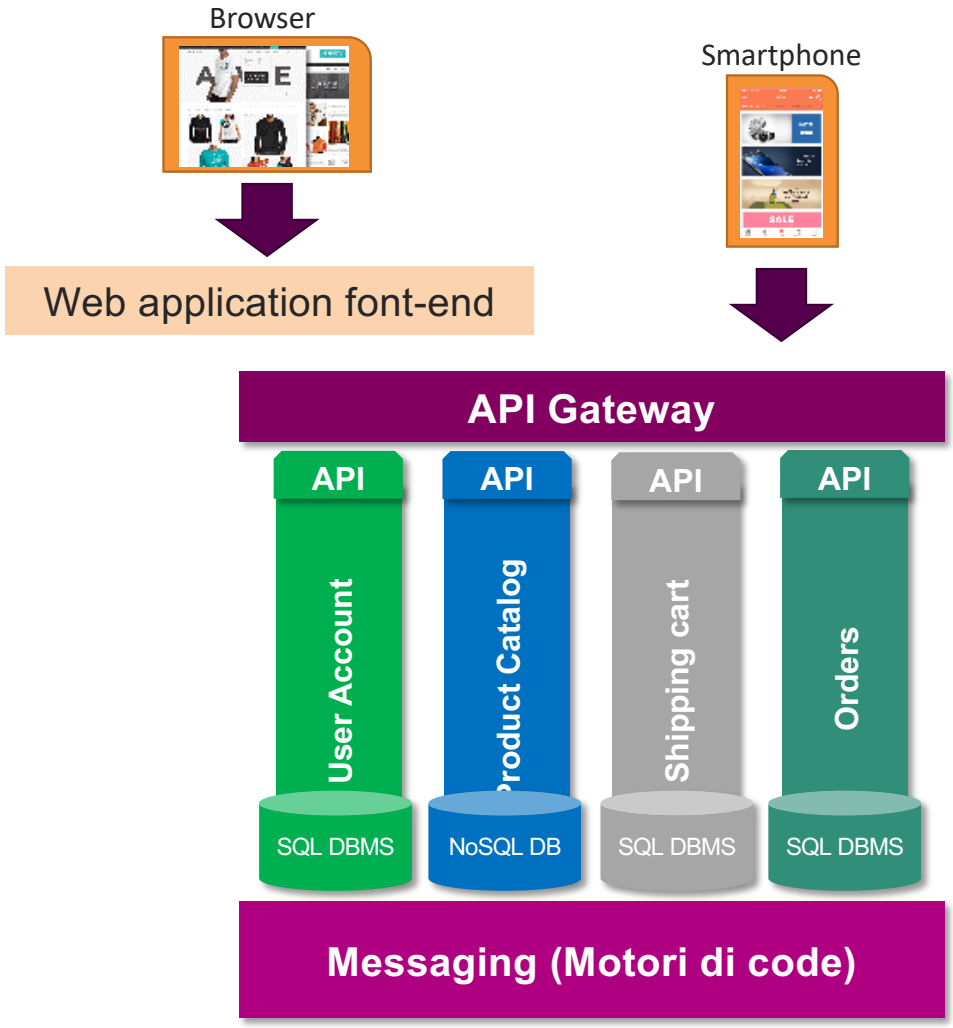
webscience an adesso company

Logical architecture: multichannel

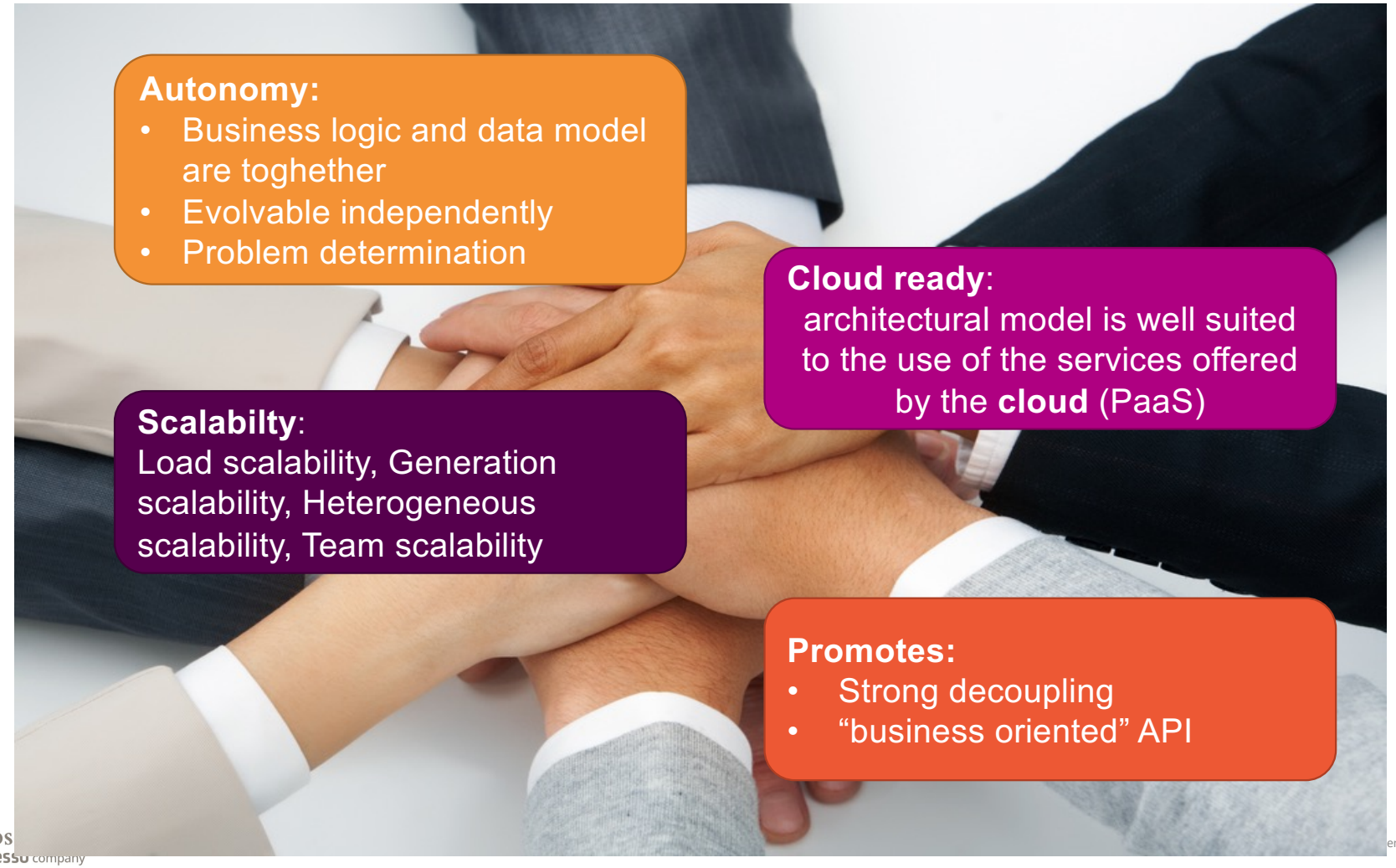
- The focus on the development of business oriented APIs enables the **reuse** of microservices in different application contexts
- The development of different **multichannel** applications is simplified



Logical architecture: multichannel - example



The strengths of microservices architecture



POLITECNICO MILANO 1863

webs an adesso company

The points of attention of microservices architecture

POLITECNICO MILANO 1863

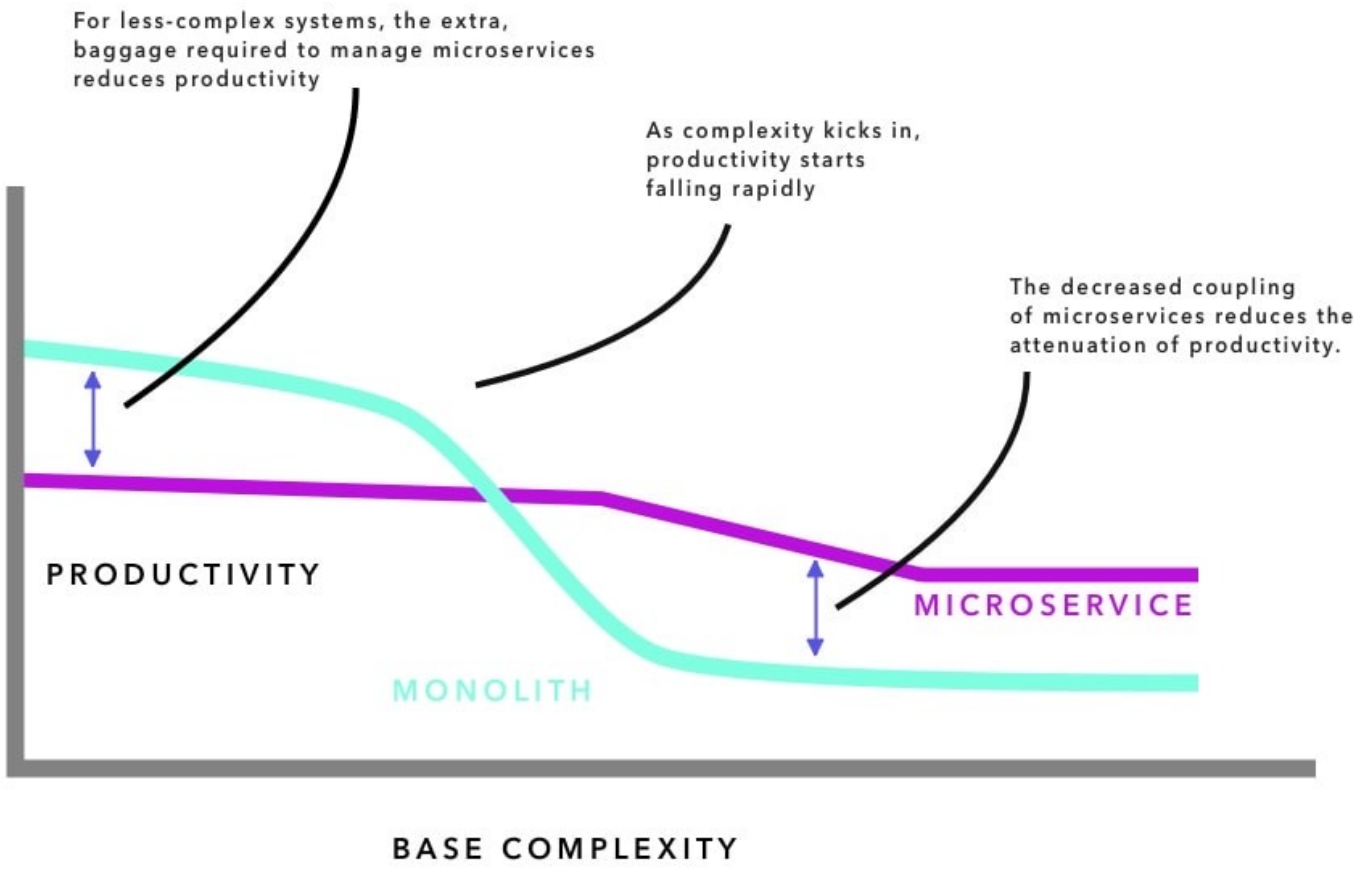
webscience an adesso company

Microservices architecture: when and how?

POLITECNICO MILANO 1863

webscience an adesso company

Monolith vs Microservices

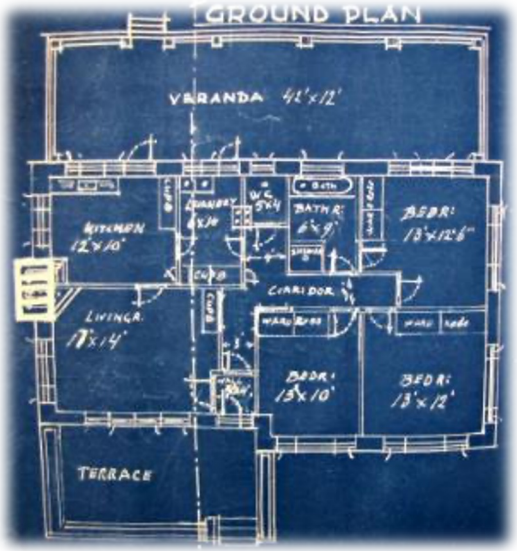


POLITECNICO MILANO 1863

webscience an adesso company

When ?

Is the monolith showing its limits ?



Do you need a better way to manage the change ?

POLITECNICO MILANO 1863

webscience an adesso company

Preconditions

Organizational:

- Are there sufficient modeling and design skills?
- Are there enough dev-ops skills?
- Is it possible to take advantage of the opportunity of PaaS products?
- Have the teams developed a sufficient culture of collaboration?
- Do the teams have sufficient experience on practices and methodologies?

Functional:

- Is the application domain (processes / entities) clear?
- Is it possible to divide the application domain into smaller functional units?
- Is it acceptable that the data model distributed among the functional units is eventual consistent?