



TORINO

#GlobalAzureTorino



Using .NET and Azure to Build & Deploy Microservices Architectures

Aka : Evolutionary Architectures

Alberto Acerbis

Scenario





The lowest level of knowledge is when we make a lot of decisions about the future system

It's easy to make things complicated, but it's complicated to make them simple

Software Architecture



The screenshot shows the Wikipedia page for "Software architecture". The page includes a search bar at the top, a table of contents on the left, and the main article text on the right. The article text discusses the definition of software architecture, its role in system design, and its application in various contexts like space shuttles and software development.

Contents [hide]

- (Top)
- Scope
- Characteristics
- Motivation
- History
- Architecture activities
 - Architecture supporting activities
- Software architecture topics
 - Software architecture description
 - Architecture description languages
 - Architecture viewpoints
 - Architecture frameworks
 - Architectural styles and patterns
 - Software architecture and agile development
 - Software architecture erosion
 - Software architecture recovery

Software architecture

Article [Talk](#)

From Wikipedia, the free encyclopedia

Software architecture is the set of structures needed to reason about a [software system](#) and the discipline of creating such structures and systems. Each structure comprises software elements, relations among them, and properties of both elements and relations.^{[1][2]}

The *architecture* of a software system is a metaphor, analogous to the [architecture](#) of a building.^[3] It functions as the blueprints for the system and the development project, which [project management](#) can later use to extrapolate the tasks necessary to be executed by the teams and people involved.

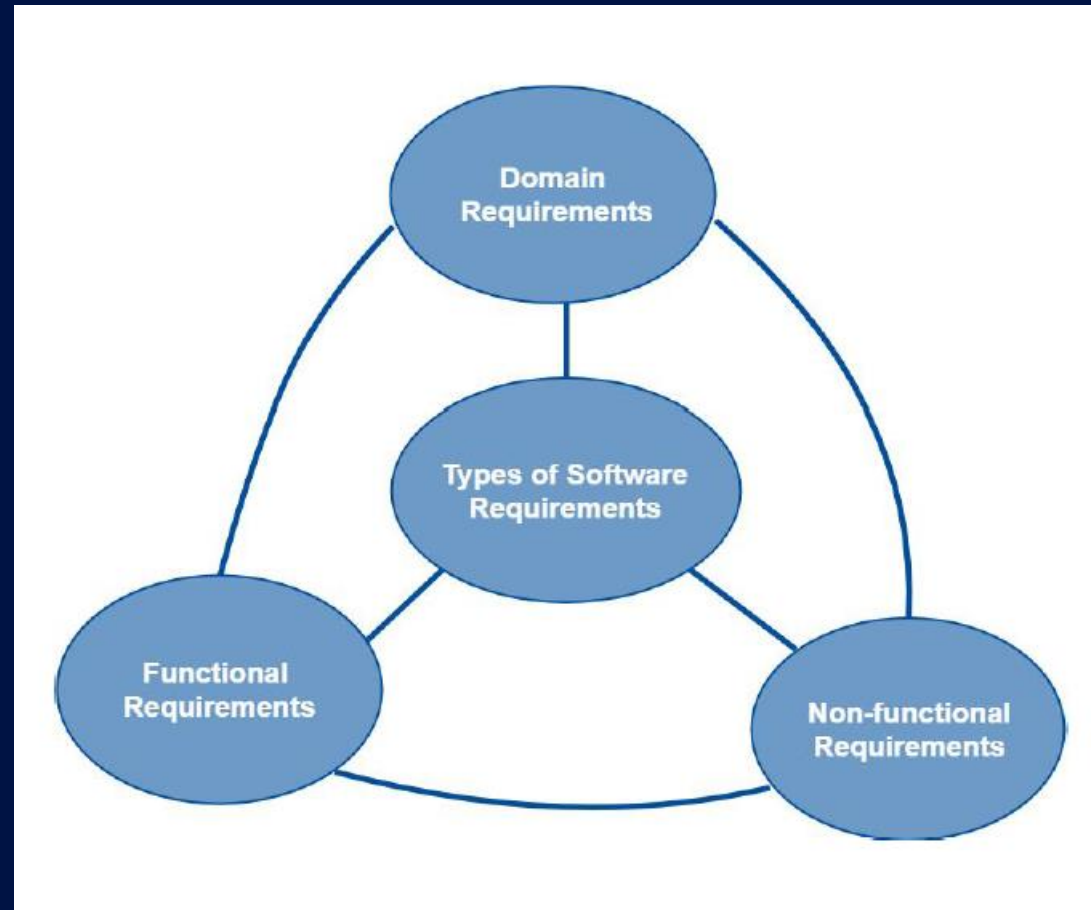
Software architecture is about making fundamental structural choices that are costly to change once implemented. Software architecture choices include specific structural options from possibilities in [the design of the software](#).

For example, the systems that controlled the [Space Shuttle](#) launch vehicle had the requirement of being very fast and very reliable. Therefore, an appropriate [real-time computing](#) language would need to be chosen. Additionally, to satisfy the need for reliability the choice could be made to have multiple redundant and independently produced copies of the program, and to run these copies on independent hardware while cross-checking results.

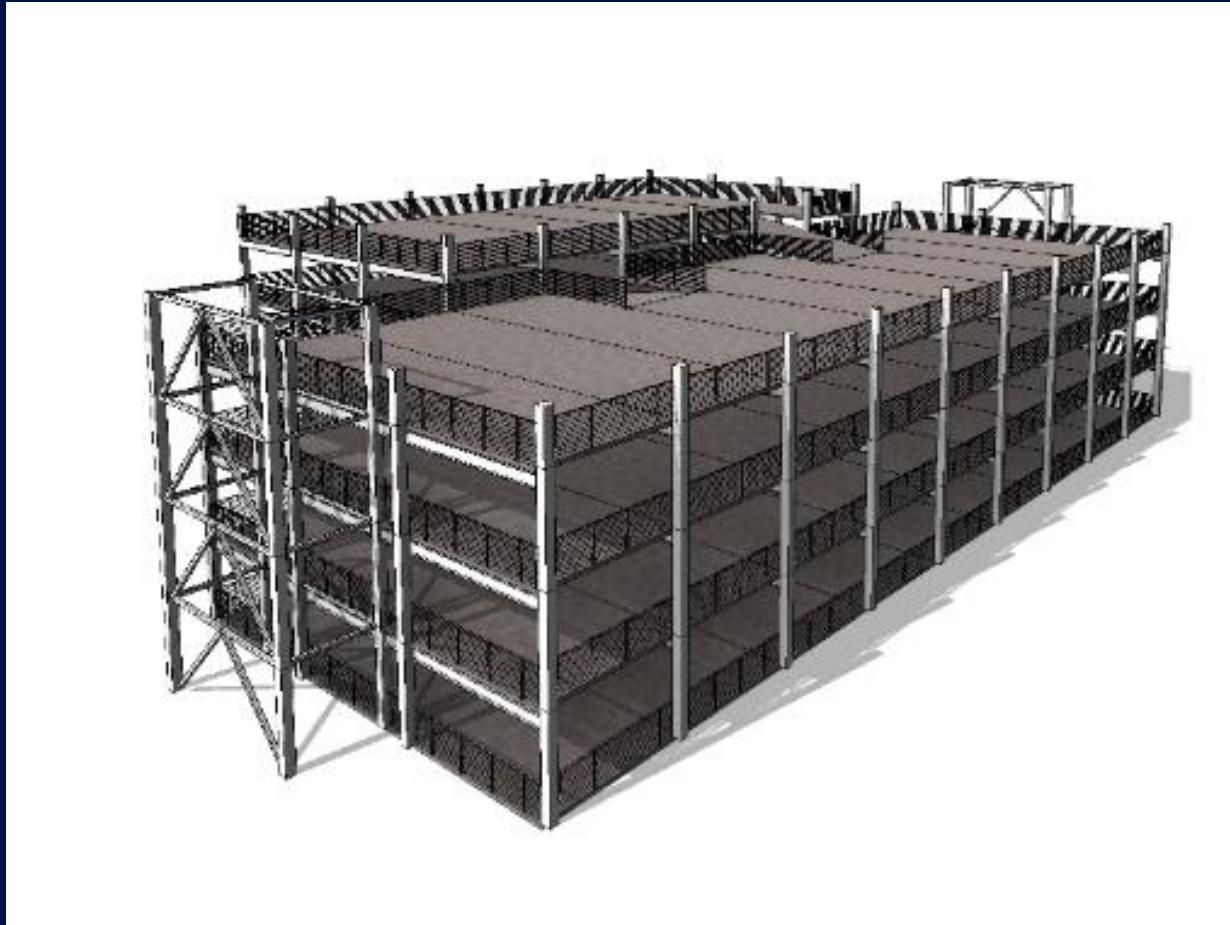
[Documenting software](#) architecture facilitates communication between [stakeholders](#), captures early decisions about the high-level design, and allows reuse of design components between projects.^{[4]:29–35}

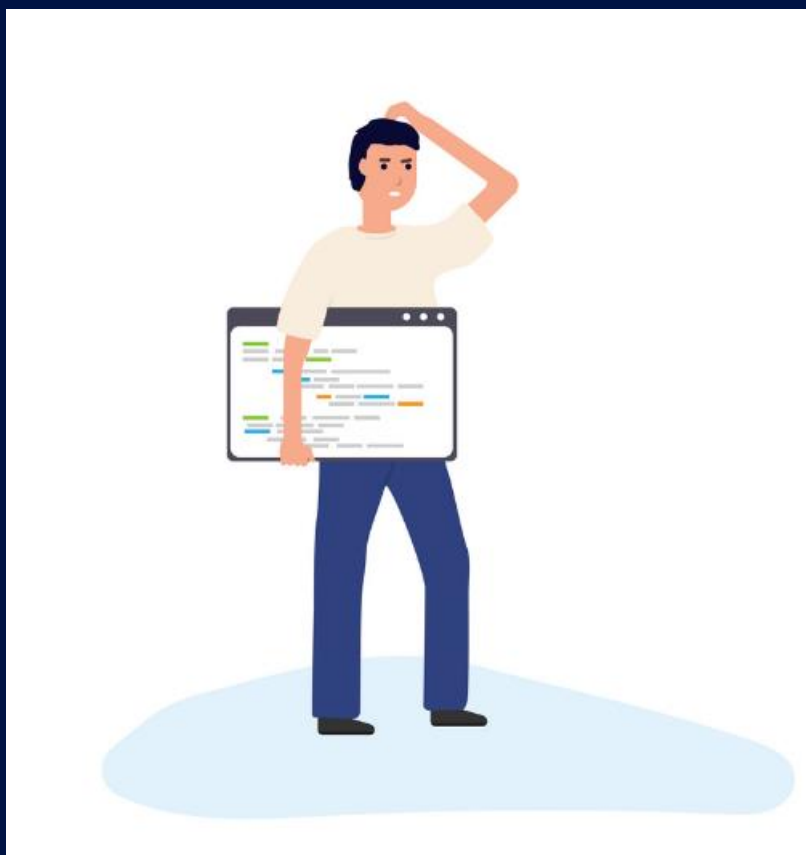
Software Architecture

- Functional requirements, aka Business Requirements
- Non-functional requirements, aka Technical Requirements
- Domain requirements, aka Industry Standard Requirements



One Upon a Time





Risk comes from not
knowing what you are
doing

(Warren Buffet)

The only way to
decrease ignorance is
to increase
understanding

New Vision

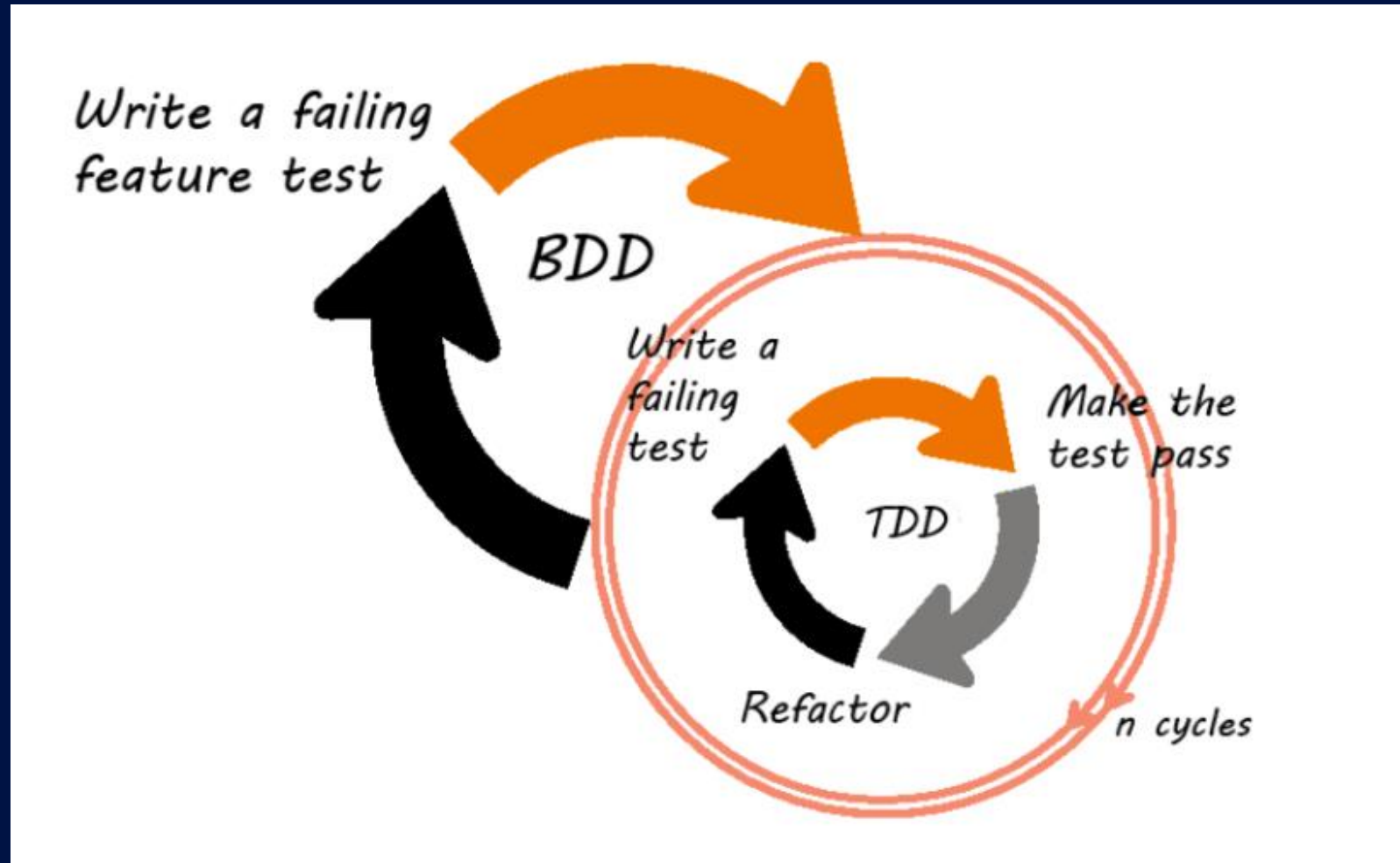
- Regular changes to the requirements have become part of normal business evolution
- From architectures as an up-front challenge to a continuous challenge
- Changes keep coming so development never ends and the so architecture will keep changing

Software Rot

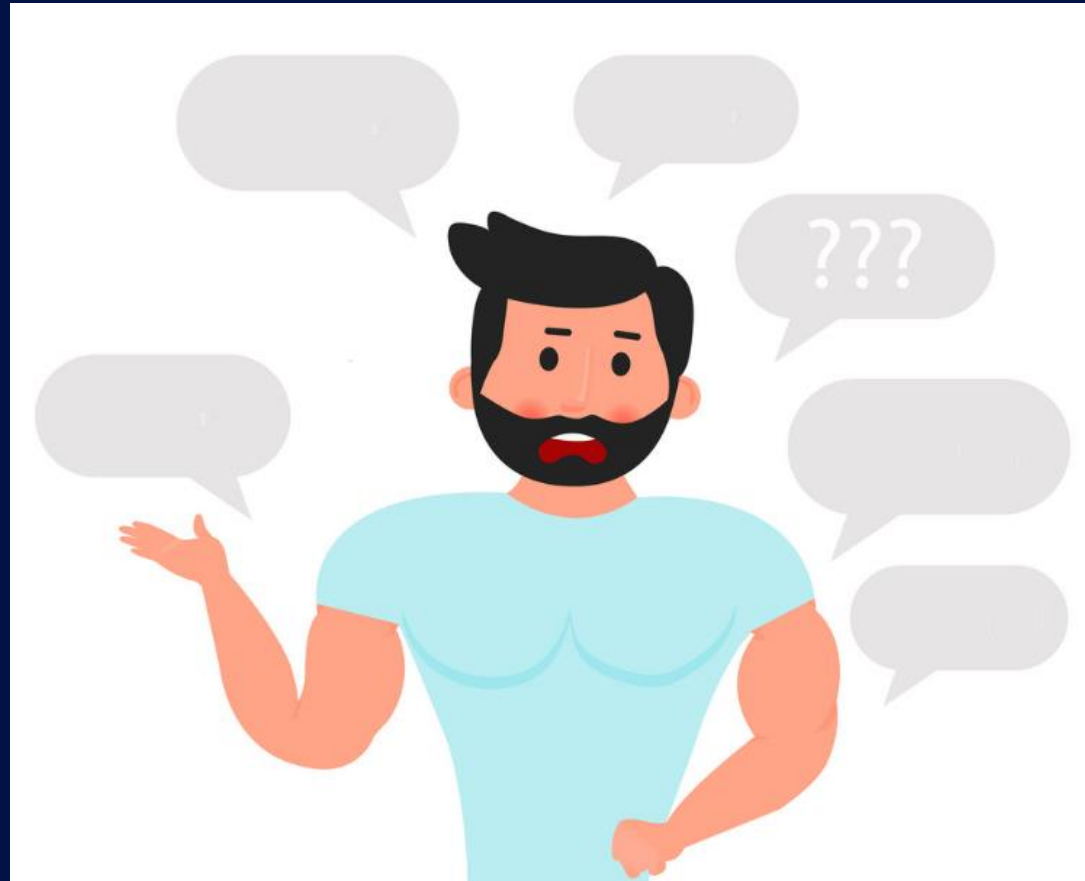
Is either a slow deterioration of software quality over time or its diminishing responsiveness that will eventually lead to software becoming faulty, unusable, or in need of upgrade.

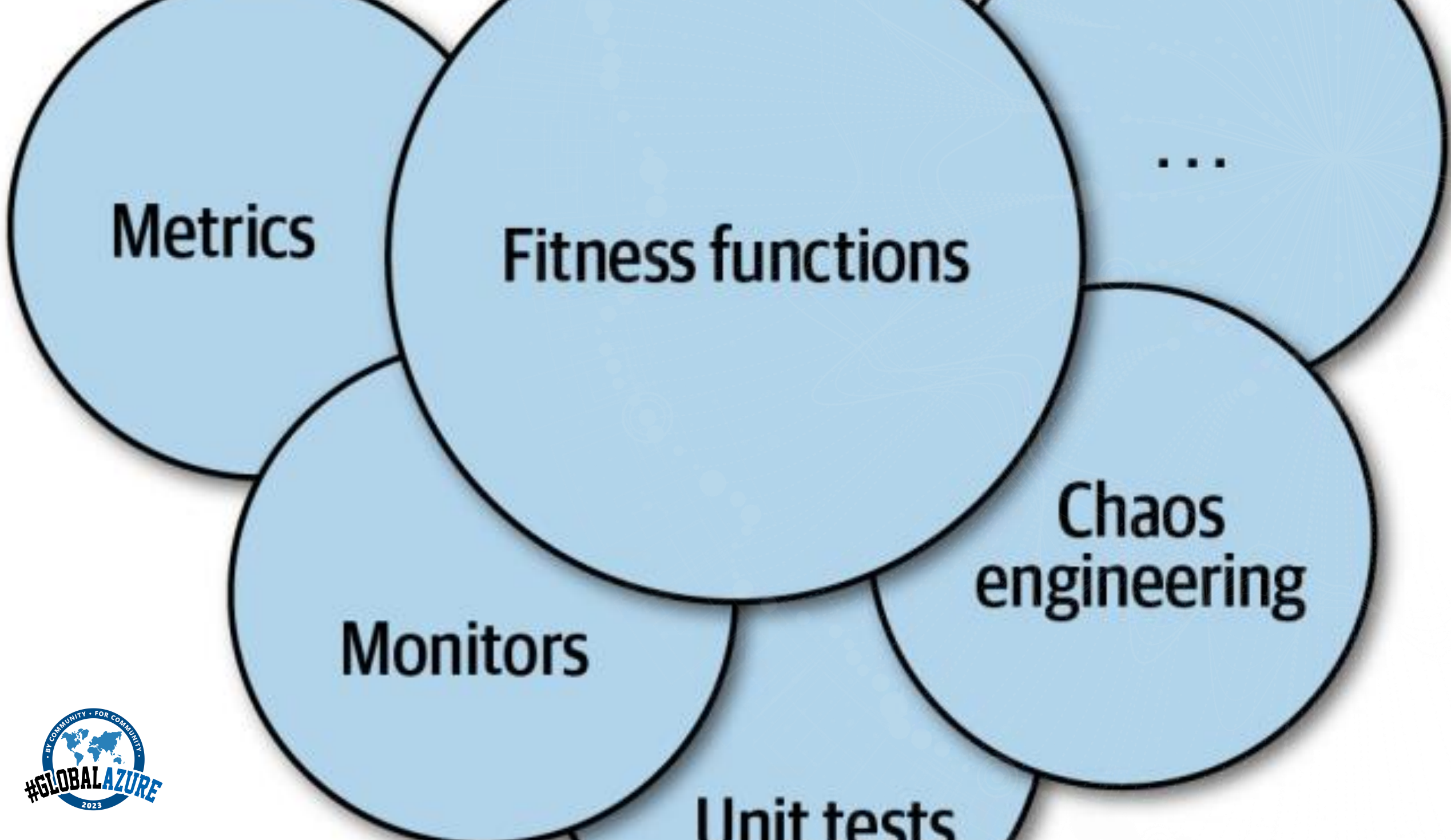


Evolutionary Software

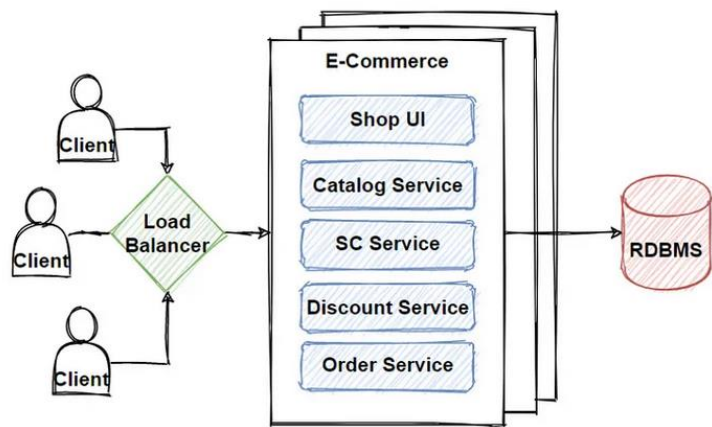


Evolutionary Architecture

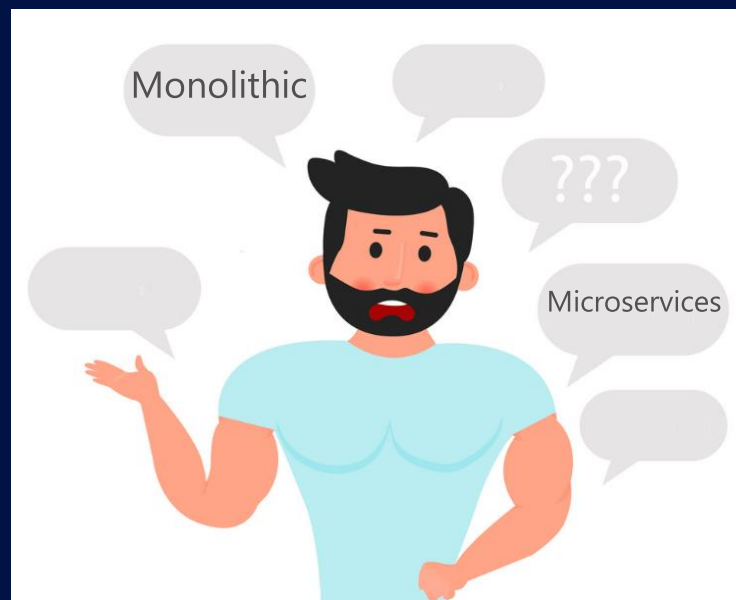
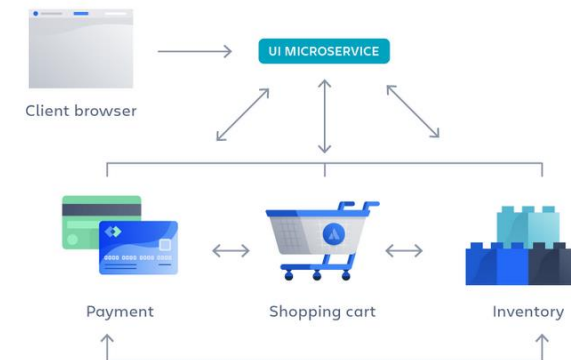




Monolithic Architecture

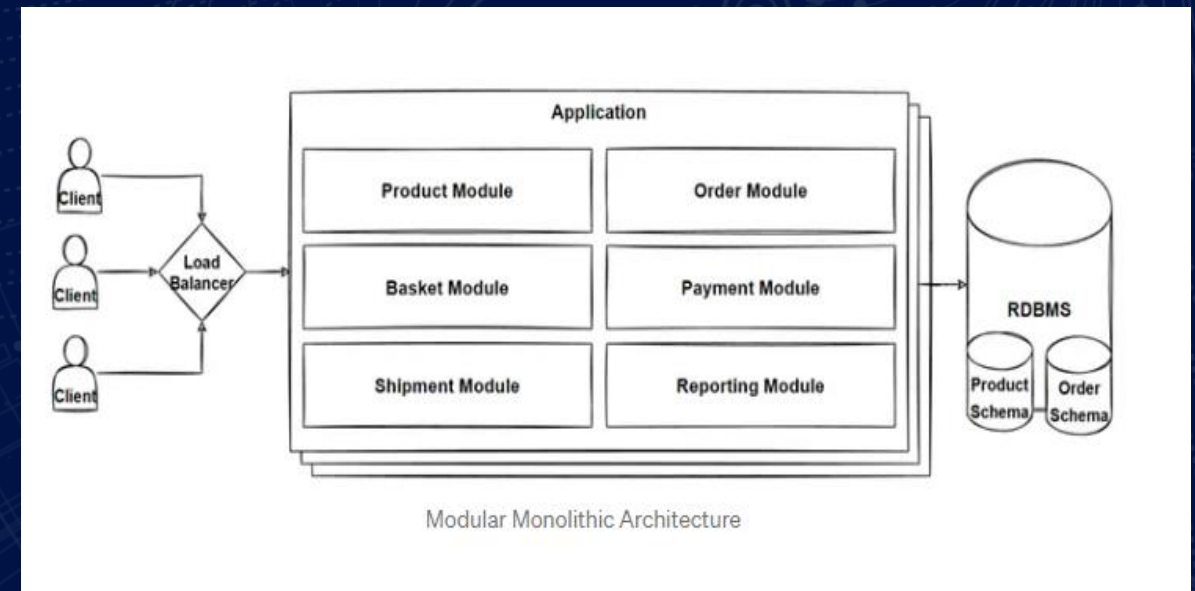


Microservice architecture

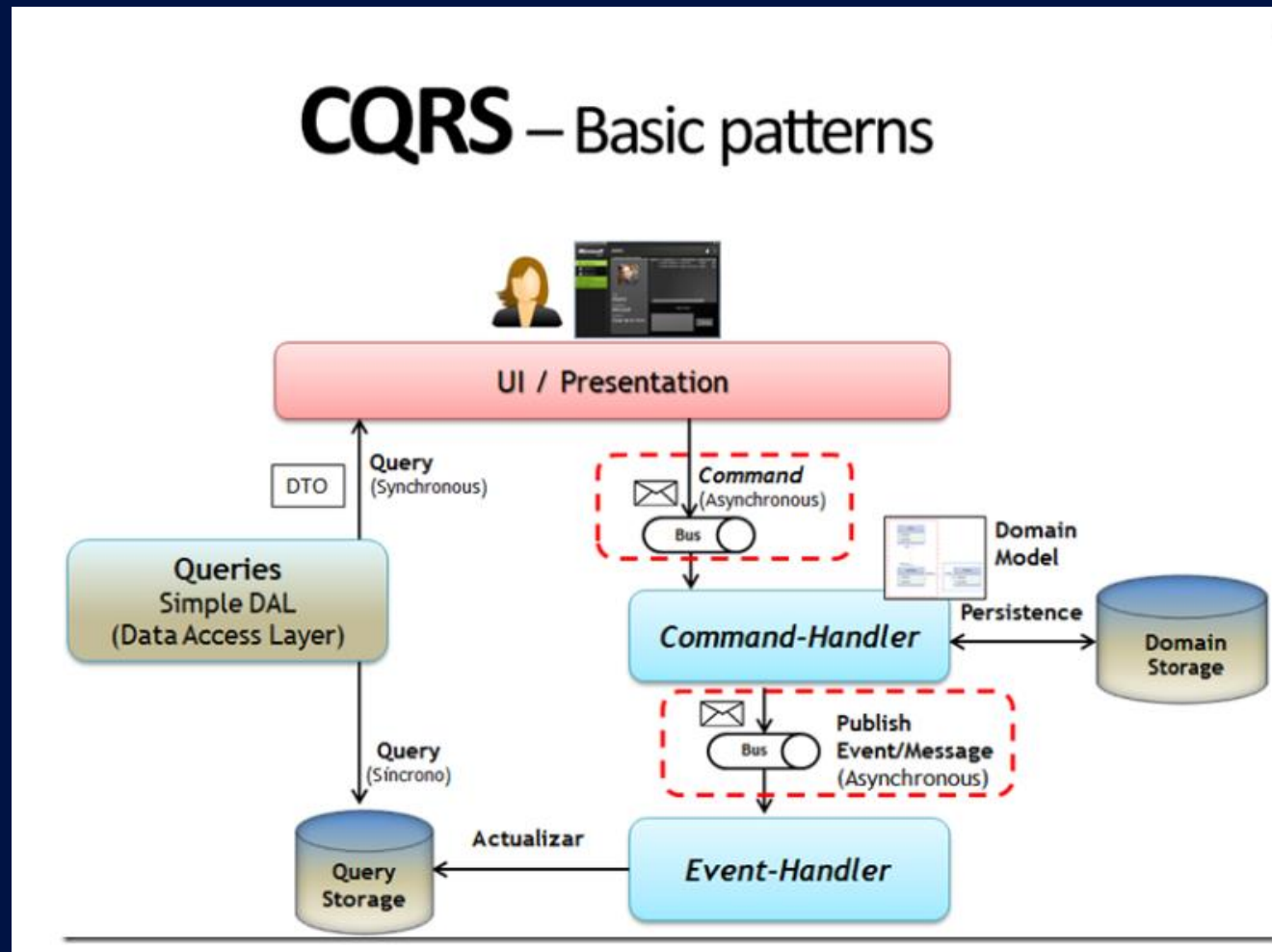


Modular Monolith Architecture

- Increased productivity
- Ease of creating new modules
- Simplicity of debugging
- Ability to convert to microservices
- Improved team collaborations
- A less complex architecture compared to microservices
- A cohesive and loosely coupled system
- The option to use different databases



Event-Driven and CQRS Pattern



[Microsoft DevBlogs](#)

Sales

Warehouse

Launch New Order

Ask Beers Availability

Beers Availability Checked

{CommandHandler}

{EventHandler}

Beers Withdrawan

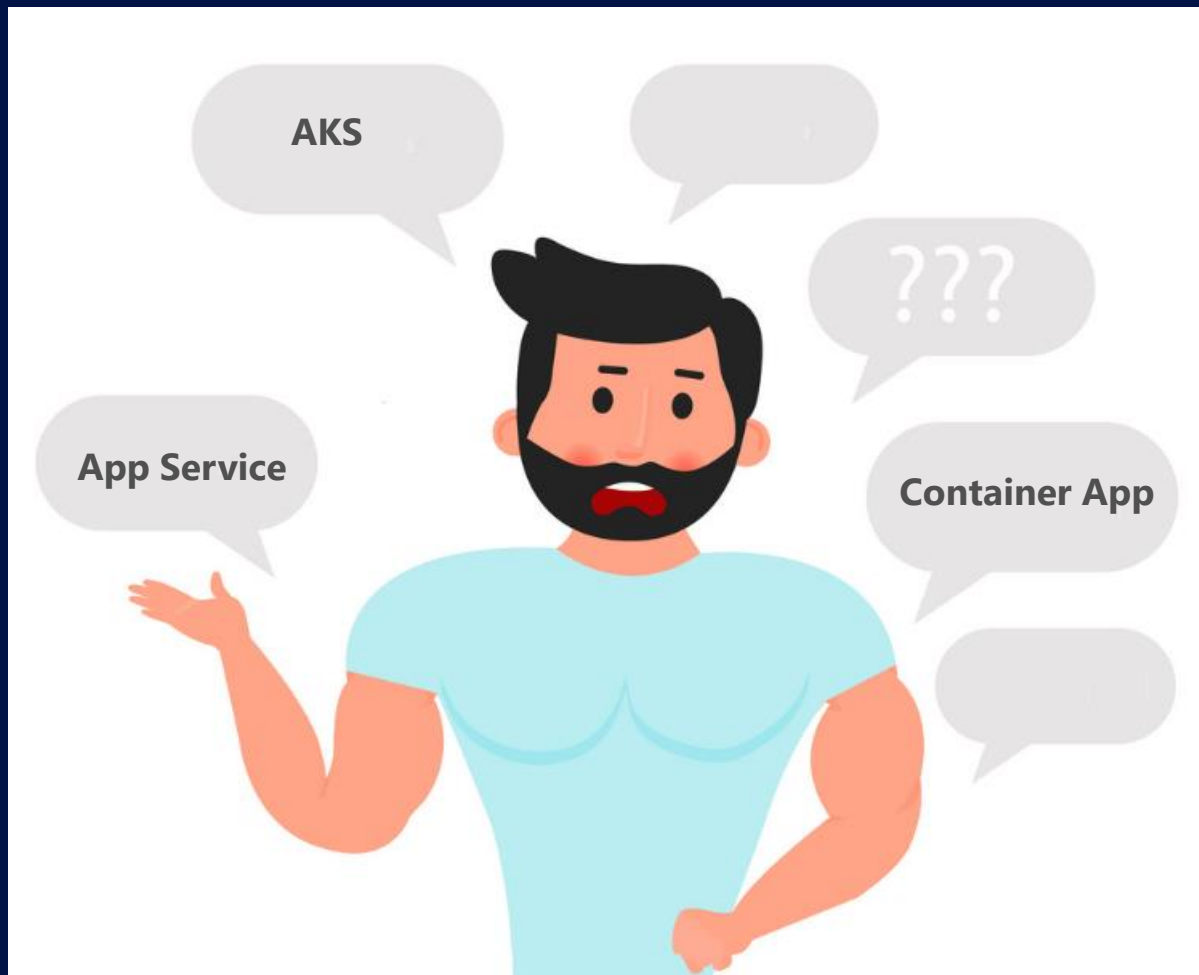
Beers Withdrawal

Add Order



mo





Azure AppService provides fully managed hosting for web applications including websites and web APIs

Azure Kubernetes Service (AKS) provides a fully managed Kubernetes option in Azure

Azure Container Apps enables you to build serverless services based on containers

Demo





Prima Legge dell'Informatica



**Se funziona
quanto basta,
non lo toccare o
si guasta**

Demo




A matter of cost




Container App




Registry




Aks per cluster




Load Balancer rules




Monitoring




Registry



Ingresses

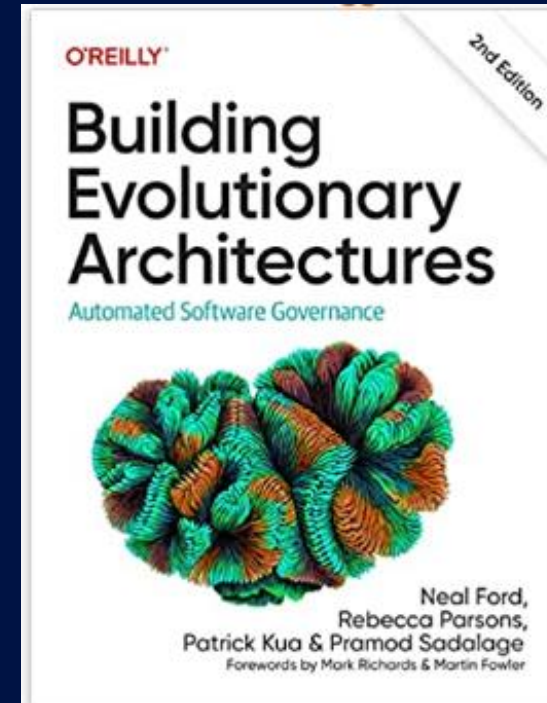
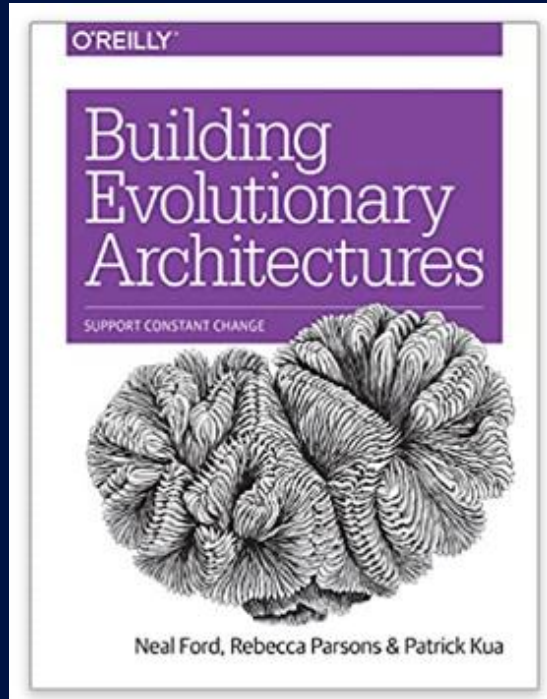


Instances



Service bus

Evolutionary Architecture



About Me

#GlobalAzureTorino



alberto.acerbis@intre.it



<https://github.com/brewup>



<https://github.com/cqrs-muflone>



<https://github.com/ace68>



<https://www.twitch.tv/dddbrewup>



alberto acerbis



#GlobalAzure



TORINO