A warm welcome to all of you.

- Thank you for attending the **DevSecOps Days Pittsburgh 2022.**
- Shoutout to the Software Engineering Institute at Carnegie Mellon University for organizing this event.
- I hope you all have a wonderful session !

DevSecOps in the Cloud from the Lens of a Well-Architected Framework

A Brief about me

Currently , an Assistant Director, Cloud Practice at EY (Ernst & Young).

Before that I led the CCOE (Cloud Centre Of Excellence Team) at Accenture.

Have 10+ years of IT experience , working on
Cloud technologies since 2017.

Expectation management -

- This is not an introductory session on how to get started with DevSecOps
- It is expected that the audience for this session is already familiar with the concept of DevSecOps , at a high-level.
- This talk is primarily targeted at individuals/teams/enterprises who want to understand how architecture plays a crucial role in DevSecOps principles, practices and patterns, and how to relate one with the other.

Agenda for today-

Over the next **25-30 minutes**, I will briefly touch on

- Section I : Context and background
- Section II : What is DevSecOps
- Section III : Architecture concepts
- Section IV : Link between DevSecOps and architecture
- Section V : Conclusion


LET'S GET STARTED!
makeameme.org

- **Section I : Context and background**
- Section II : What is DevSecOps
- Section III : Architecture concepts
- Section IV : Link between DevSecOps and architecture
- Section V : Conclusion

Let's start with some context

From physical machines to VM to containers, we have really **come a long way.**

Earlier, our focus was on **on-prem systems**, from there we went on a modernization journey to **Cloud-based workloads**, and now the talk of the town is **Cloud-native architectures.**

But, do you know what has **stayed the same**, amidst all this technology innovations and changes?

The business requirements have not changed.

The business **does not care** about **the latest technology toolchain or innovation that we are consuming**.

They are only concerned about their **business outcomes**, nothing else.

So, while new frameworks and technologies will obviously act as a catalyst, and unlock potential opportunities, but at the same time, **the basic requirement for a well-architected secure workload that can help serve the business use-case stays the same.**

Sometimes, we engineers are **unduly influenced by new technology, and toolkits**, which obviously bring a lot of value to the ecosystem, but we should not, at the same time **ignore/undervalue the essence of it all – Architecture**.

The requirements for scalable, **secure**, fault-tolerant, performant workloads or systems are **not new**, and have not come up only because of the cloud-native revolution.

Those requirements were always there, and no matter what paradigm or technology framework or toolchain we practice/implement, we must always ensure that **we do not compromise on the basic architectural tenets.**

But, this leads to some questions

But, we were doing fine till now.

Why did we need to pivot to this new way of working ?

Why do we need to embrace this so-called **DevSecOps** ?

Will it enable us to fulfil our original business requirement of having a well-architected, secure workload that can server our user needs?

To be able to answer this question, we need to understand,

- What is DevSecOps
- What is Architecture
- How can architecture play an important role in DevSecOps

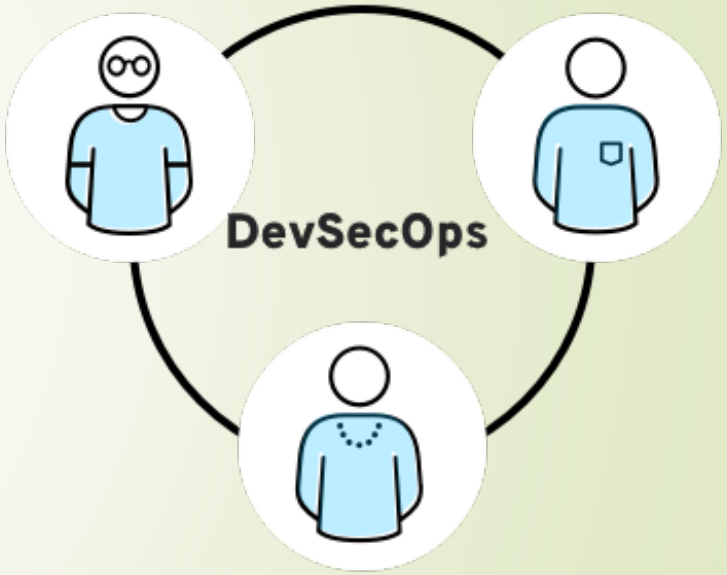
- Section I : Context and background
- **Section II : What is DevSecOps**
- Section III : Architecture concepts
- Section IV : Link between DevSecOps and architecture
- Section V : Conclusion

DevSecOps stands for development, security, and operations.

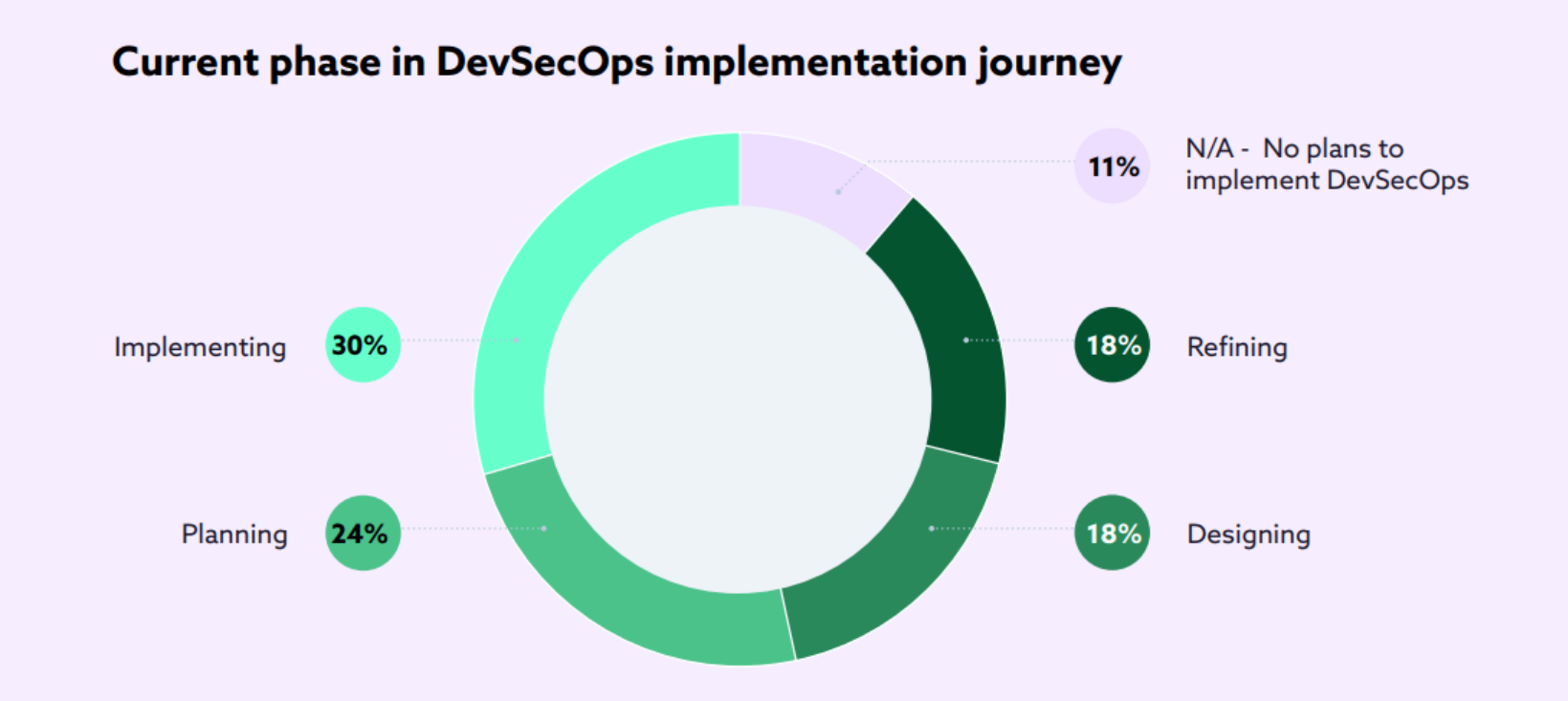
It's an approach to culture, automation, and platform design that integrates security as a **shared responsibility throughout the entire IT lifecycle**.

Primary aspect of DevSecOps

- **Security** is a **shared responsibility** integrated from end to end.
- Thinking about application and infrastructure security **from the start**.
- Automating security gates to keep the DevOps workflow from slowing down.



Current stage in **DevSecOps adoption** across our enterprise landscape



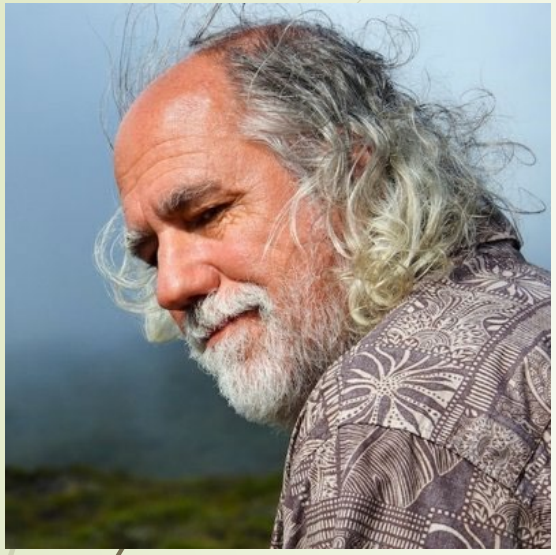
As per Secure DevOps and Misconfigurations Report, 2021 by Cloud Security Alliance

- Section I : Context and background
- Section II : What is DevSecOps
- **Section III : Architecture concepts**
- Section IV : Link between DevSecOps and architecture
- Section V : Conclusion

What is architecture?

IEEE 1471:2000 definition -

''The fundamental organization
of a system embodied
in its components, their
relationships to each other,
and to the environment, and the
principles guiding
its design and evolution.''



Grady Booch , IBM Fellow

"All architecture is design but not all design is architecture; architecture represents the set of **significant design decisions** that shape the structure and behaviour of a system where **significance is measured by cost of change**."

Ref - https://twitter.com/grady_booch/status/1257355498598342656?lang=en

What is architecture ?

Martin Fowler

"the set of **design decisions that must be made early**"

Ref -https://www.youtube.com/watch?v=DngAZyWMGR0

Matthew Parker ,
previously Global Head of engineering – Pivotal Labs

"Architecture , in the field of software development ,
are **decisions that are hard to reverse** "

Some examples could be,

"Whether to deploy our application on container based compute environment like Kubernetes or go with a serverless model like AWS Lambda "

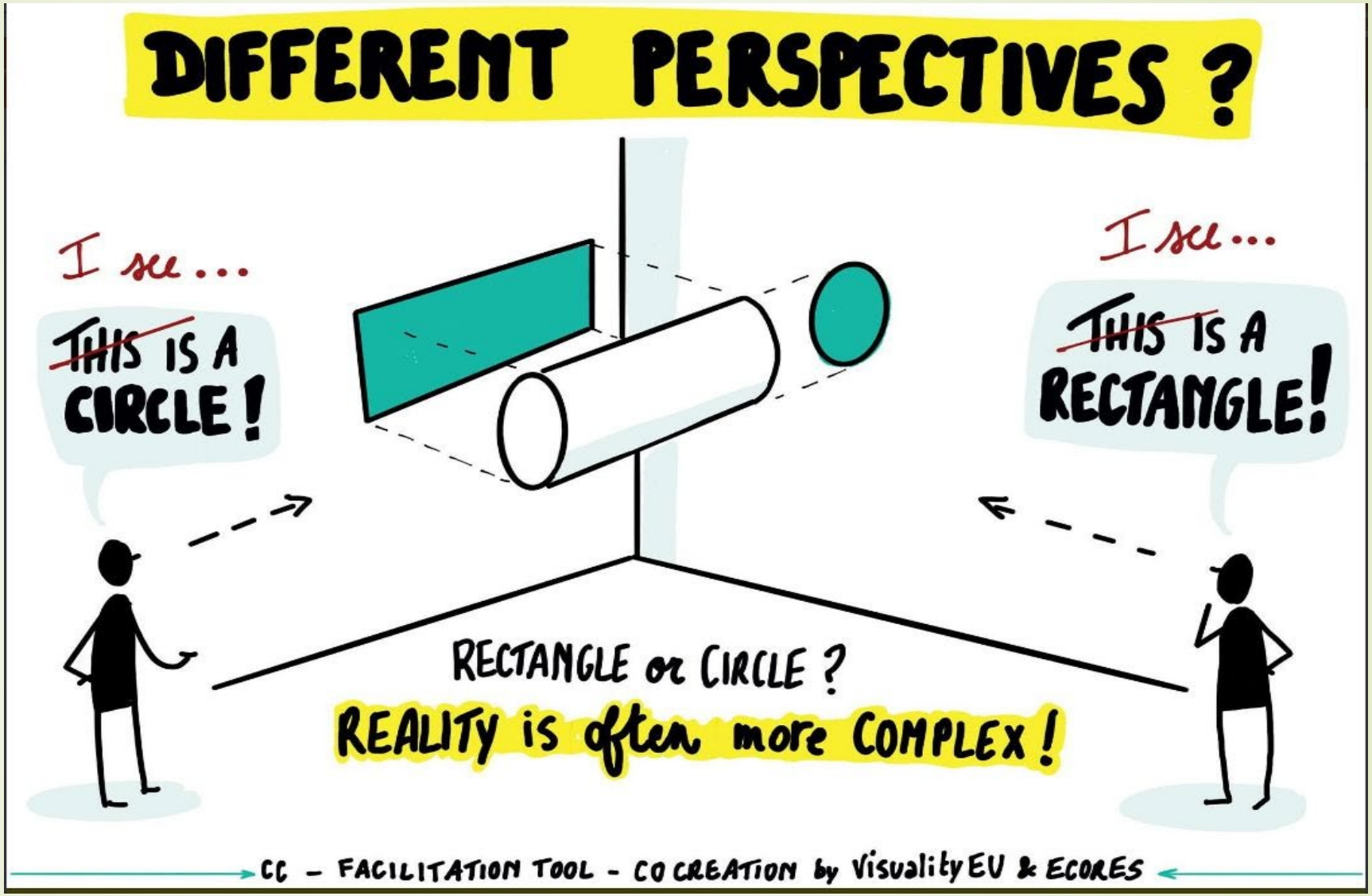
OR

"Whether to go with synchronous API-based communication or use an event-bus based asynchronous messaging system like RabbitMQ/AWS SQS"

OR

"What programming language do you use for your application development"

Ok , then who/what are architects ?


DIFFERENT PERSPECTIVES ?
I see...
~~THIS~~ IS A CIRCLE!
I see...
~~THIS~~ IS A RECTANGLE!
RECTANGLE or CIRCLE ?
REALITY is often more COMPLEX!
CC - FACILITATION TOOL - CO CREATION by Visuality EU & ECORES

"Architects have a **holistic understanding** of the system , across a different spectrum of views and viewpoints , and they offer a **different perspective** of the system/design."

Gregor Hohpe ,
Enterprise Strategist , AWS

"The architect doesn't have to be the smartest person in the room. Instead, they make everyone else smarter."

https://architectelevator.com/architecture/multiple-dimensions/

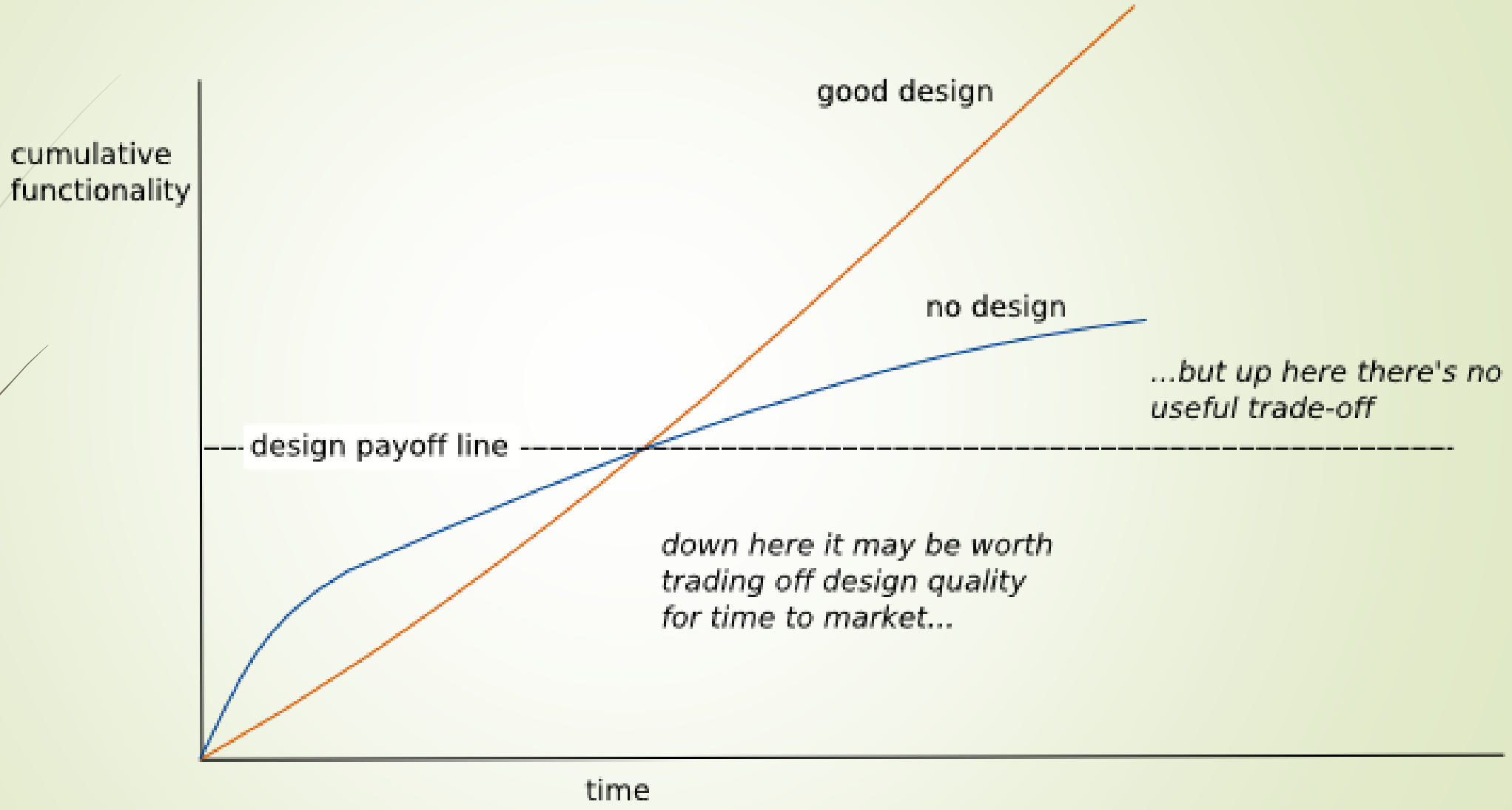
But why do we need architecture ?


If you think good architecture is expensive, try bad architecture.
— Brian Foote —
AZ QUOTES

Architecture matters -

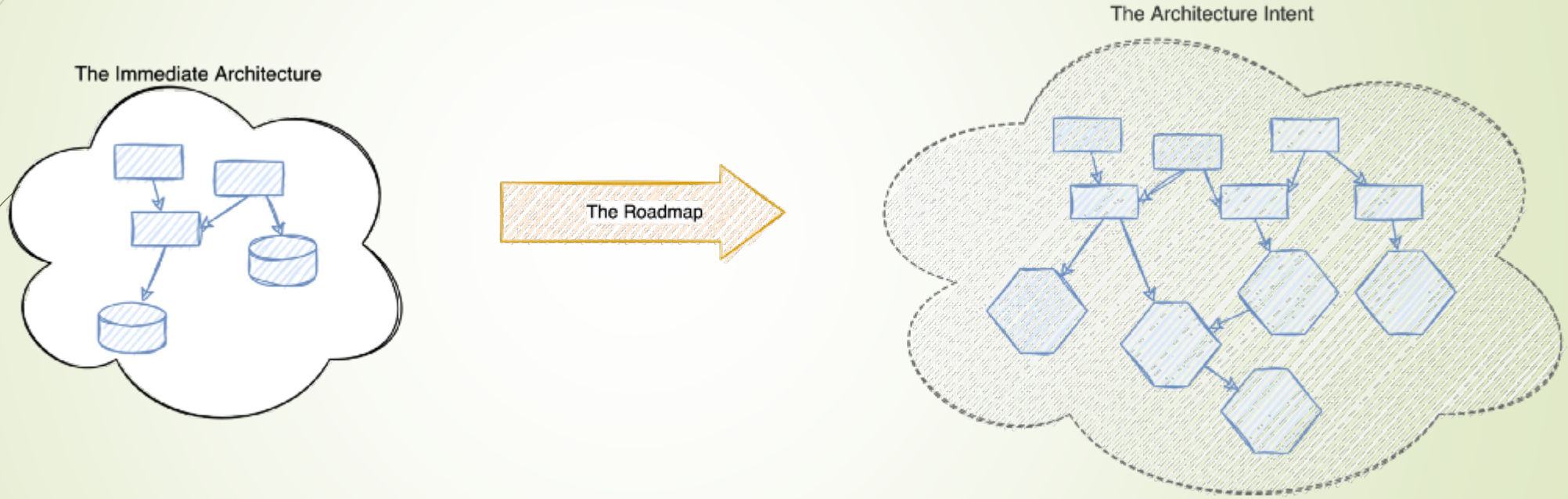
"Bad architecture / no architecture **slows down** the ability of our customers to compete , as **over time it becomes harder and harder to ship new features**"

Let me explain ,



Ref - https://martinfowler.com/bliki/DesignStaminaHypothesis.html

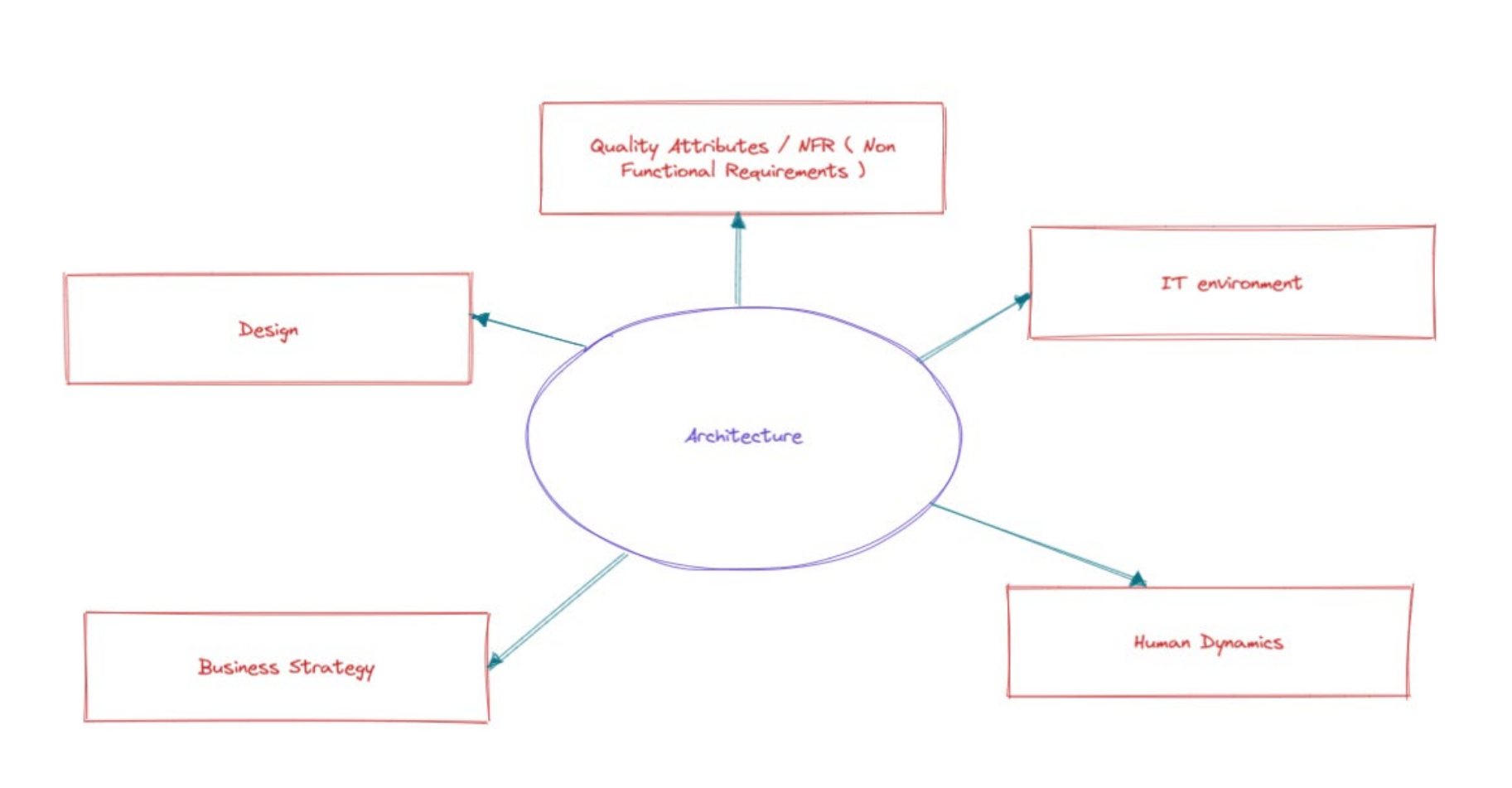
Agile architecture - Immediate Architecture to Architecture Intent



Ref - https://betterprogramming.pub/agile-architecture-intent-vs-the-immediate-36d327289b2d

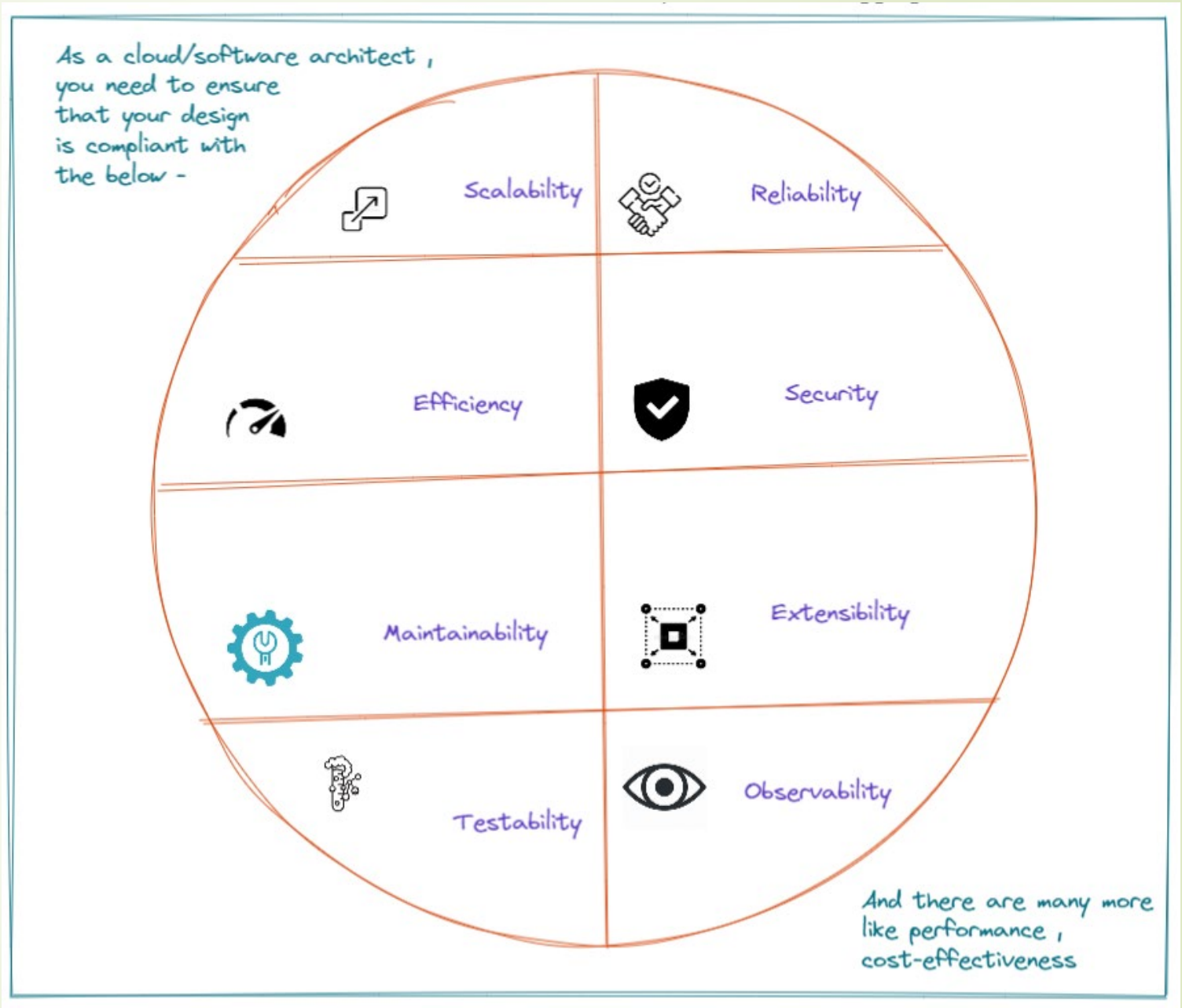
Not having a design based on a strong architectural foundation will ultimately lead to not being able to achieve business objectives.

Quite frankly , architecture is extremely complicated topic.



But if we really need to start the architectural journey, it is easier to get started with quantifiable parameters that cab be measured and enhanced.

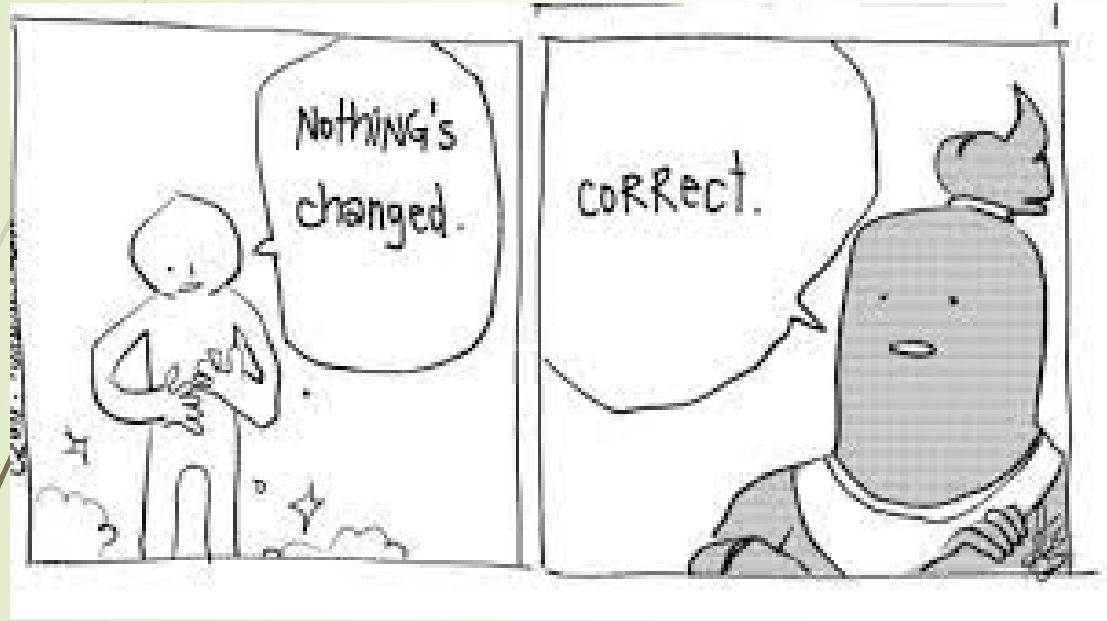
This brings us to NFR(s) or Non-Functional Requirements



As you might agree - the cloud might be new , but the **essence of architecture remains the same** .

- Section I : Context and background
- Section II : What is DevSecOps
- Section III : What is architecture
- **Section IV : Link between DevSecOps and architecture**
- Section V : Conclusion

Did you notice something ?



Essentially , even in the new world of cloud/cloud-native,
nothing has changed that much .

All the constraints / quality attributes that you needed to maintain in your datacentre based applications **are relevant even now , more so than ever** .

Moving / migrating to the cloud , does not mean that you do not need to think about **Security** anymore .

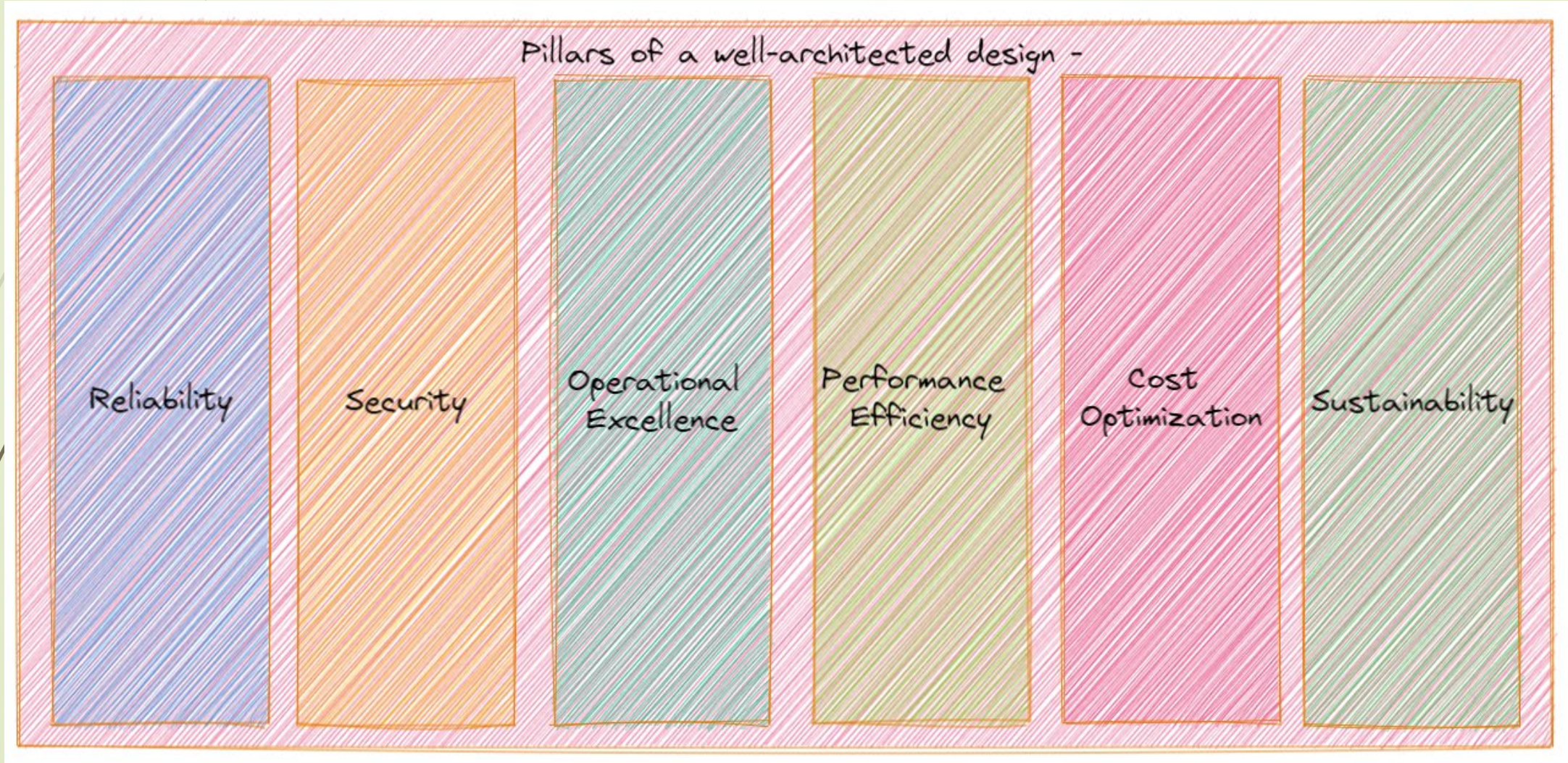
The same constraints are present in the cloud or cloud-native ecosystem also ,
but the toolchain available is more robust and easy-to-consume.

Well-architected frameworks

- At its core, a well-architected framework is a **set of best practices that help organisations optimise workloads.**

- While following the principles laid out in a cloud provider's well-architected framework aren't necessarily requirements, they do provide some of the best available advice for building an architecture that best manages security, cost, performance, and reliability.

For example, let us look at the AWS Well-Architected Framework -



From the outset, it might look like Security is just one of the many pillars of the WAF, but if you look closely you will find that **security pillar in WAF is not just related to the security NFR , rather related to other NFRs like reliability, performance, deployability, and so on.**

Let's take an example –

One of the primary activities in DevSecOps/Security Left initiatives is to integrate some sort of SAST tools either in the IDE/CI pipeline.

If this was not there, there would be a chance that some vulnerabilities will remain in the final product that might be exploited in production.

Say, we do not follow a DevSecOps approach and the vulnerability is only discovered in production either by testers/ by malicious actors who are trying to exploit it.

At that time, we might need to patch the code, or libraries to mitigate the vulnerability which might even result in application downtime.

And if the application goes down, **we are actually impacting another NFR - Reliability**

Let's take another example –

Say, we do not inject security into our CICD pipelines, or we do not shift-left in terms of security, in that case, our security testing by InfoSec teams will only happen at the end of the deployment, maybe before the production release.

In that case, say our security testing happens once every 6 weeks, then if a potential vulnerability is identified/exposed at that stage, then the production deployment will be stalled/postponed.

So, basically my security flaw is impacting my ability to deploy production-ready increments of code/features to my business users.

So, we can indirectly imply that – a lack of DevSecOps practices is resulting in the application not being "deployable" enough.

This is basically another very important NFR in software design/architecture, called '**Deployability**'

"At a high-level , if you really think about it ,

DevSecOps principles

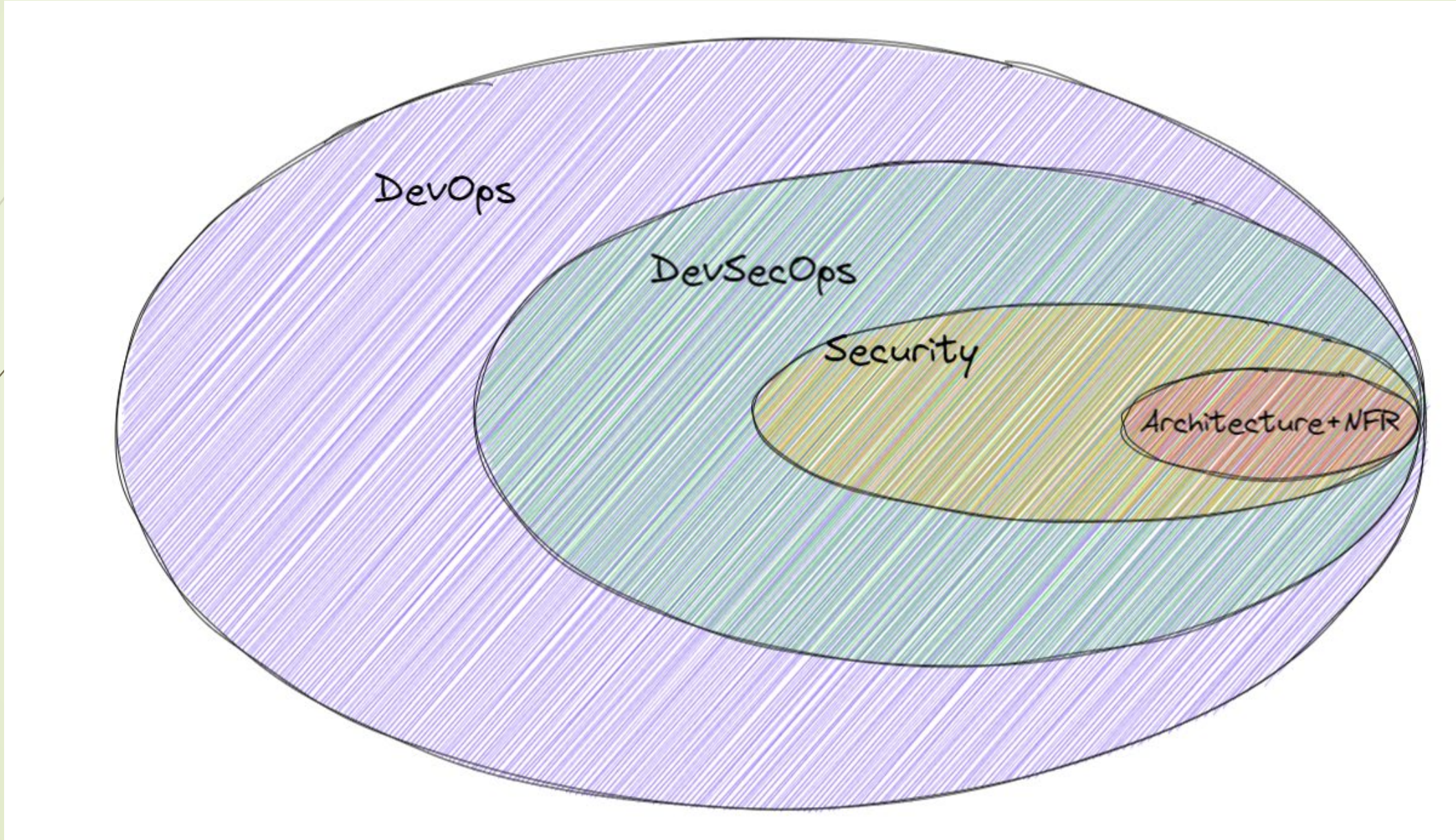
are **extensions of the best practices/standards**

that enforces the same NFR(s)

that we discussed"

- Section I : Context and background
- Section II : What is DevSecOps
- Section III : What is architecture
- Section IV : Link between DevSecOps and architecture
- **Section V : Conclusion**

Conclusion



We need to ensure that the principles, patterns and policies that we implement across an enterprise with the idea of putting in place a DevSecOps practice, **is based on the foundations of architecture.**

Today, there is DevSecOps, tomorrow there will be something more advanced and tuned to the need of the hour, but the **understanding of architecture, even at a basic level will always be important.**

Thank you so much !!