



From Legacy to Innovation: Modernizing 20-year-old apps in an Agile World

Gartner Application Innovation &
Business Solutions Summit



Amir Rapson
Co-Founder and
CTO/CCSO
vFunction



Nenad Crncec
Founder





Pioneering AI-Driven Architectural Observability

Founded in 2017 | Headquarters: Menlo Park, CA, USA



Recognized innovator, multiple patents



2023 Stratus Award for Cloud Computing



2023 Intellyx Digital Innovator Award



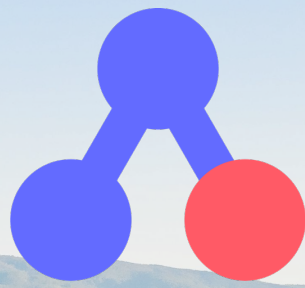
The 20 Coolest Cloud Software Companies of the 2024 Cloud 100

vFunction named a 2024 Gartner® Cool Vendor

In AI-Augmented Development and Testing for Software Engineering.

The GARTNER COOL VENDOR badge is a trademark and service mark of Gartner, Inc. and/or its affiliates, and is used herein with permission. All rights reserved. Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's Research & Advisory organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.



ARCHITTECH

Our mission is to (re)design your experience, transform business models and practices by delivering complete digital innovation and integration, making your business more efficient, adaptive to change and scalable.

ARCHITECH



At Architech, expertise is more than just a promise – it's a core value that defines who we are and drives everything we do.

PARTNERS



STRATEGY



ARCHITECTURE



TRANSFORMATION



TECHNOLOGY



AI & DATA



DELIVERY



What we do ?



STRATEGY



ARCHITECTURE



TRANSFORMATION



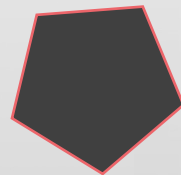
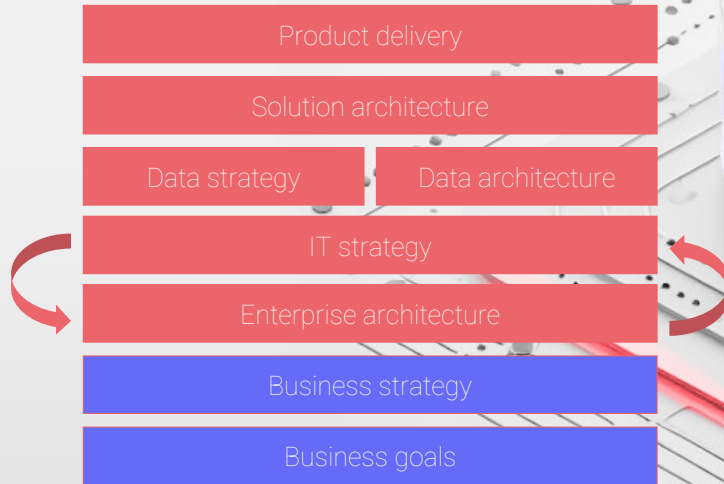
TECHNOLOGY



AI & DATA



DELIVERY





Building a structure that supports our business

Business capabilities

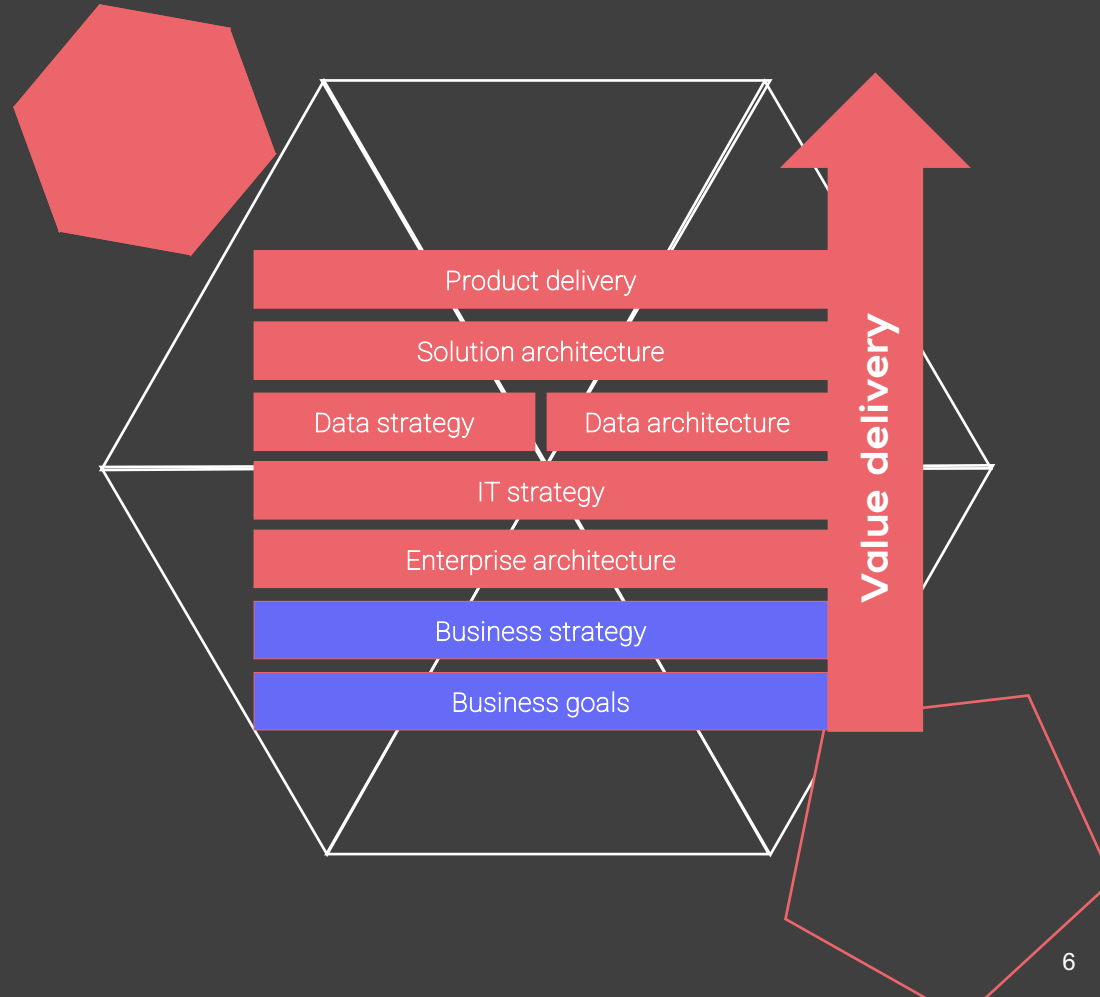
Processes

Technology

People

All-in-one solutions

Monolith systems



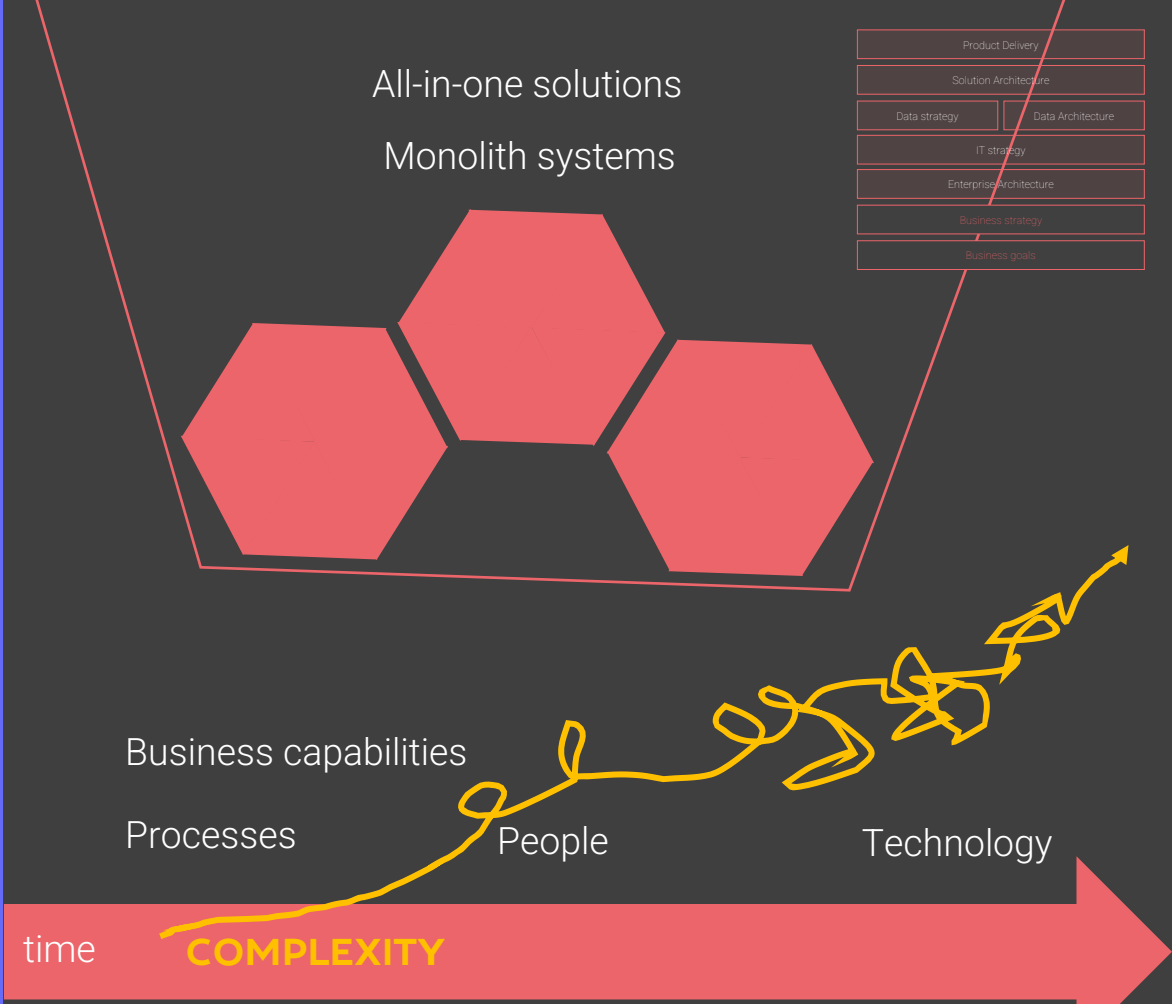


Growing the structure

Growing complexity

Customer demand

Market conditions





Characteristics

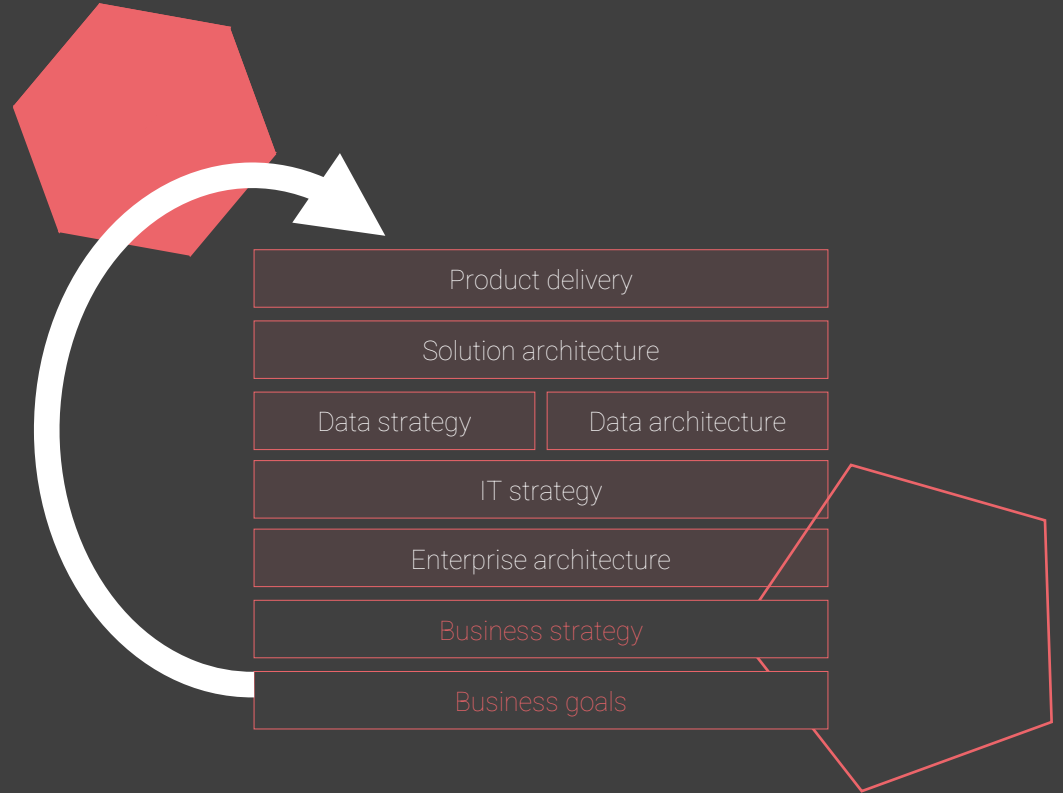
Difficult to change

Change takes time

Risk

Process bound

Unknown unknowns





What we end up doing

Quick feature delivery

Tight deadlines

No change governance

No architecture governance

Workarounds and
“temporary solutions”



Product Delivery

Solution Architecture

Data strategy

Data Architecture

IT strategy

Enterprise Architecture

Business strategy

Business goals



How do we manage changes

Modernization

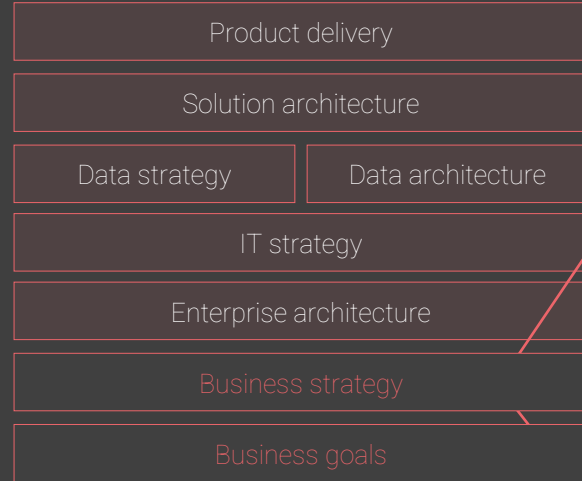
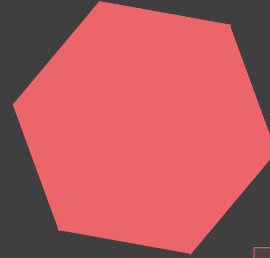
Digital transformation

Technology transformation

Agile delivery

Access to data

Knowledge management





What do we get ?

Disconnected systems

Dysfunctional

Suboptimal

Change-averse

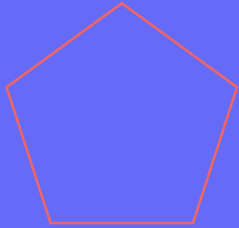
Closed & opaque

Technical debt





Drivers



- Modernization
- Digital transformation
- Technology transformation
- Agile delivery
- Access to data
- Knowledge management

User centric

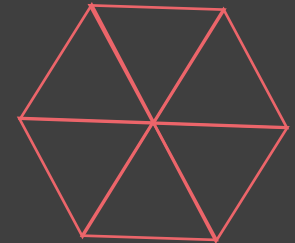
Responsive

Mobile first

Reactive

Data driven

Real-time



Customer Experience

Employee Experience

Customer Expectations

Modernization

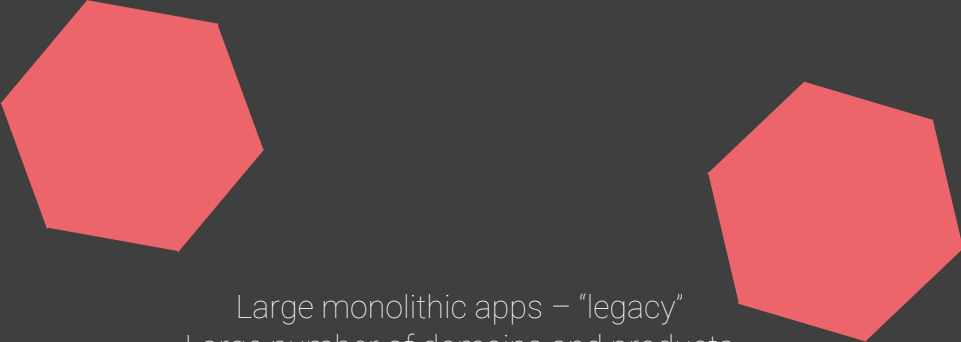
Digital transformation

Technology transformation

Agile delivery

Access to data

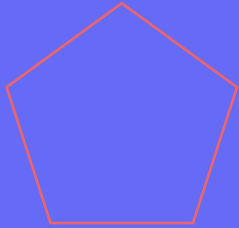
Knowledge management



Large monolithic apps – “legacy”
Large number of domains and products
Large number of cross-domain processes contained in one single runtime
Vendor lock-in
Missing “knowledge” and skills
Maturity levels
Capabilities



How to start ?



Modernization

Digital transformation

Technology transformation

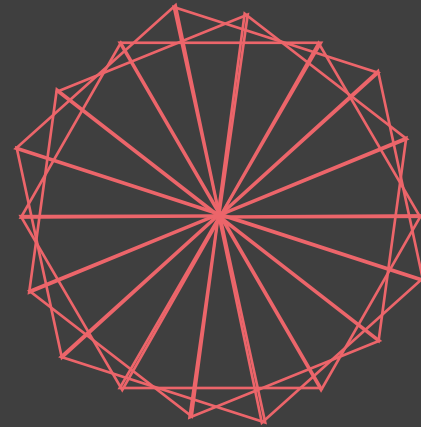
Agile delivery

Access to data

Knowledge management

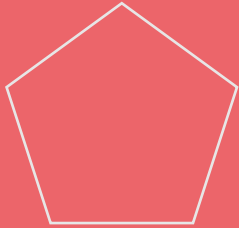
User centric Mobile first Data driven Responsive Reactive

Real-time Customer Experience Employee Experience Customer Expectations





How to start ?



Modernization

Digital transformation

Technology transformation

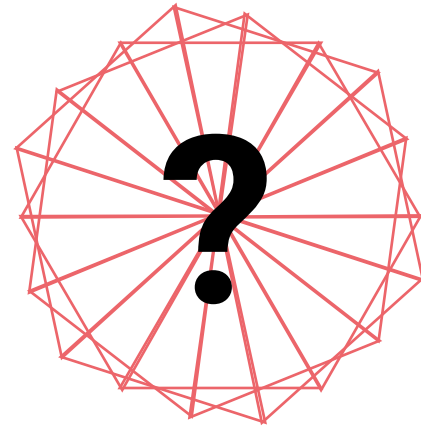
Agile delivery

Access to data

Knowledge management

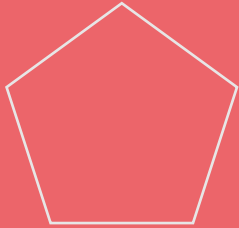
User centric Mobile first Data driven Responsive Reactive

Real-time Customer Experience Employee Experience Customer Expectations





How to start ?



Modernization

Digital transformation

Technology transformation

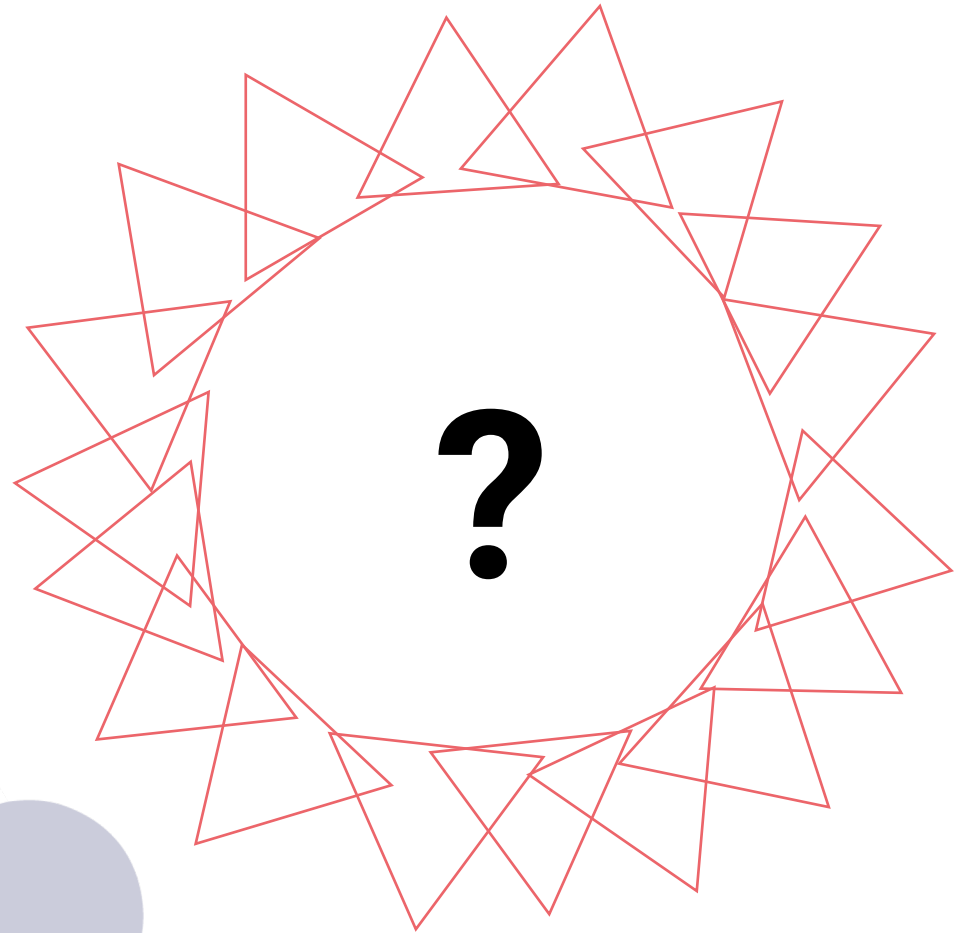
Agile delivery

Access to data

Knowledge management

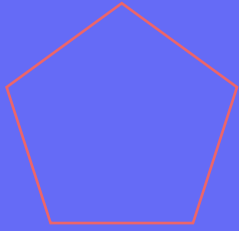
User centric Mobile first Data driven Responsive Reactive

Real-time Customer Experience Employee Experience Customer Expectations

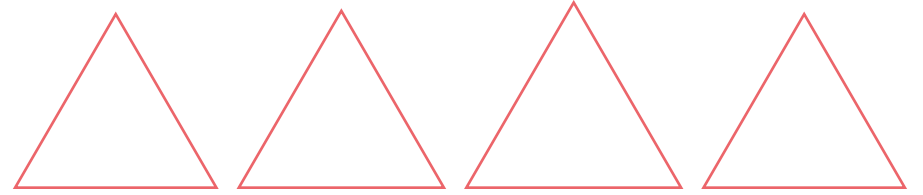
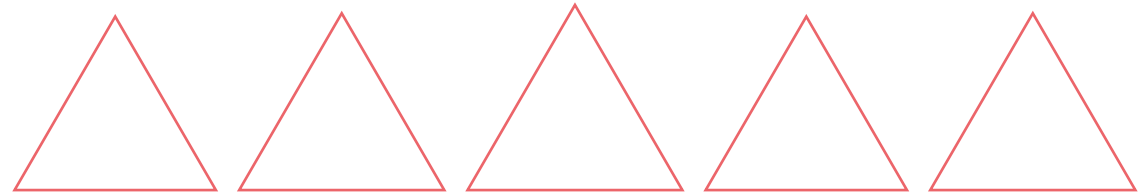
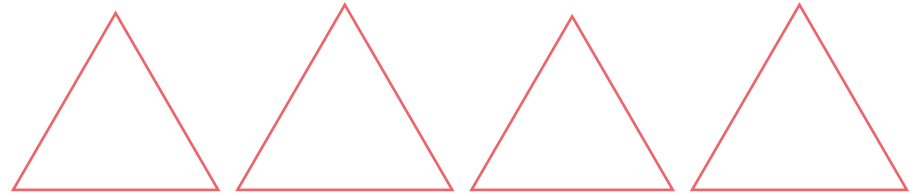
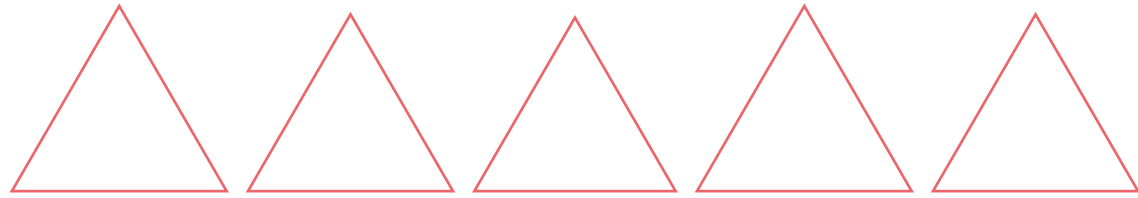




Ways to go about...



Decoupling



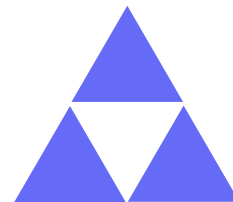
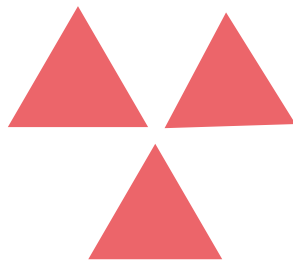


Ways to go about...



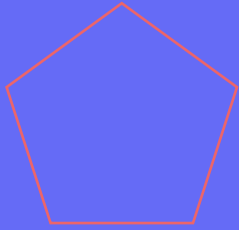
Decoupling

Domain driven design





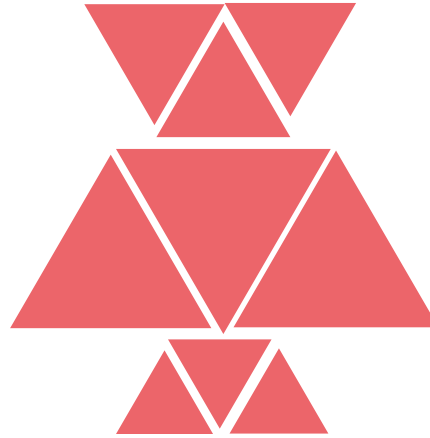
Ways to go about...



Decoupling

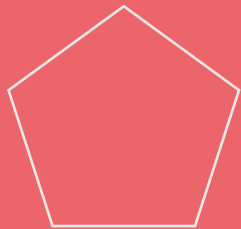
Domain driven design

Product oriented





Ways to go about...

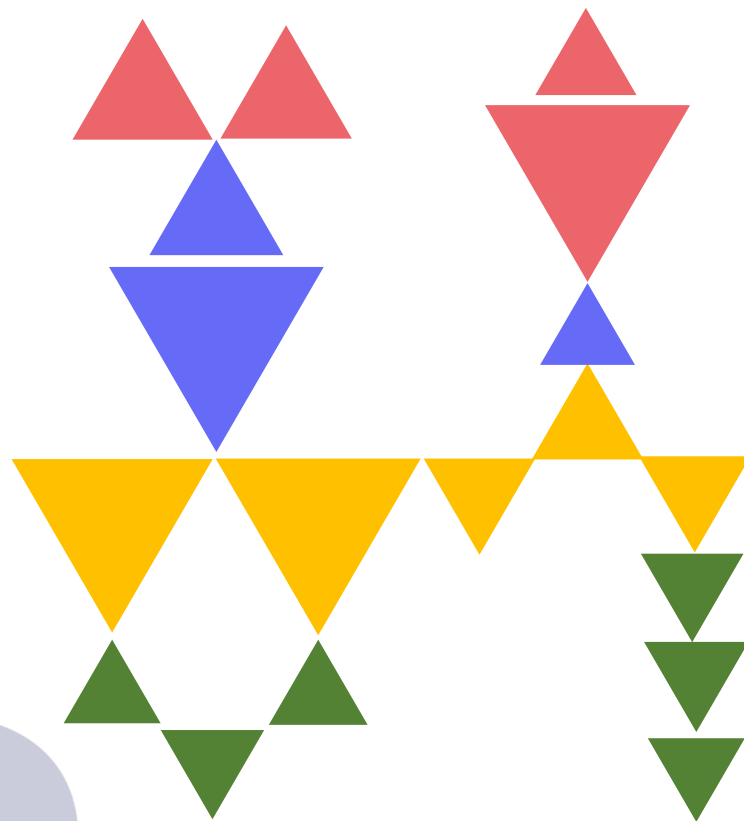


Decoupling

Domain driven design

Product oriented

Platforms





Breaking the bad architecture cycle

Knowledge management

Understanding domain

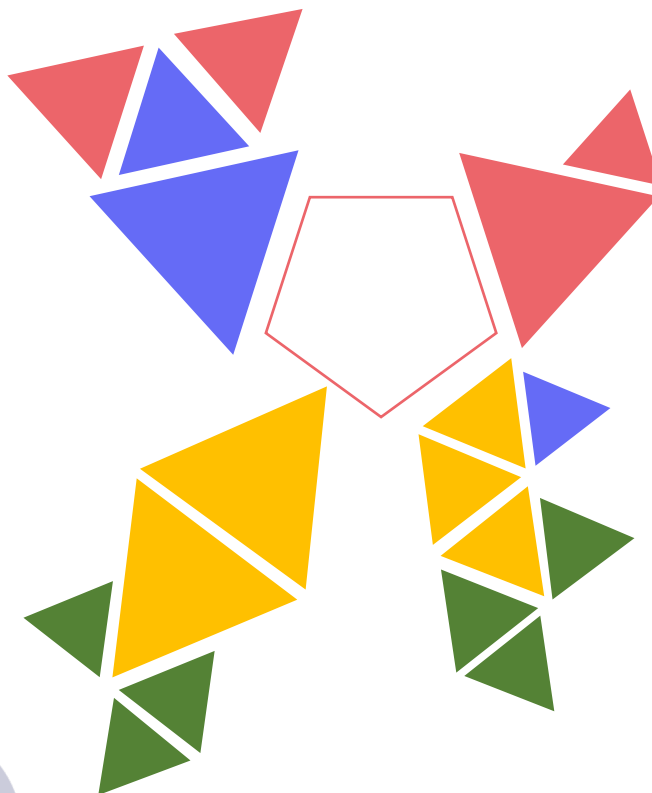
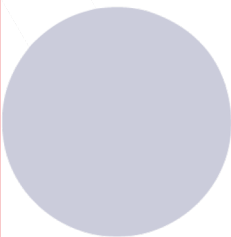
Understanding processes

Tracking changes

Transparency

Observability

Clarity



How we did it



> 10,000 employees
> 2M customers

Technology transformation

Modernisation and re-engineering

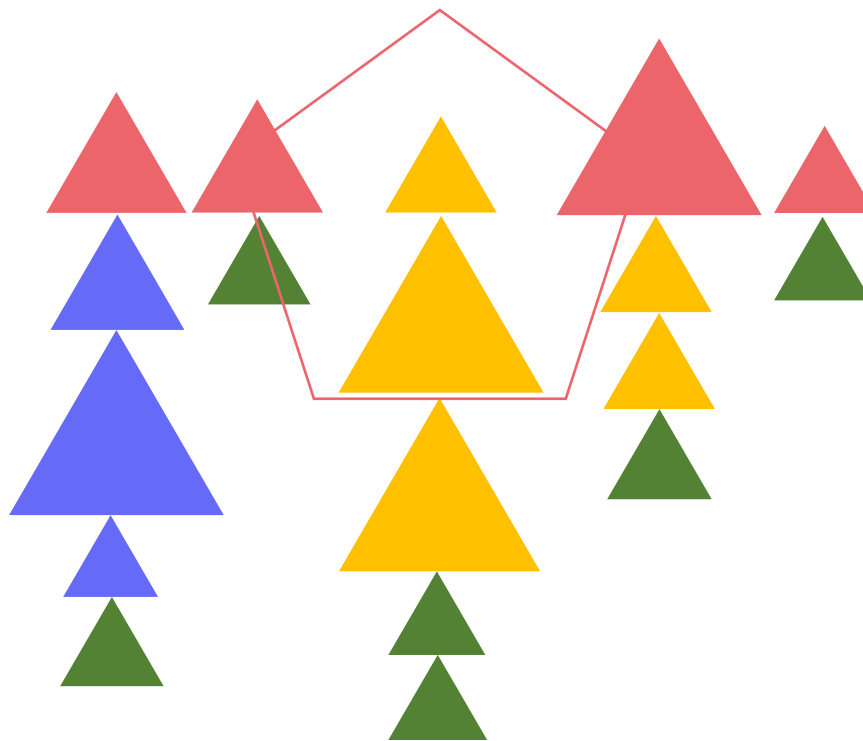
Agile transformation

Product teams

Platform teams

Integration teams

Governance





How to...

Bring common understanding to how applications & services work.

Ensure that there is no accumulation of technical debt.

Track architectural changes across deployments.

Ensure observability in distributed system.

Re-engineering required domain & solution knowledge
Lack of proper documentation
Workforce fluctuation meant knowledge is lost

Lack of delivery quality
Production issues
Poor UX
Prolonged delivery deadlines



What Architech brings

Architecture governance

Target architecture and
Target operating models

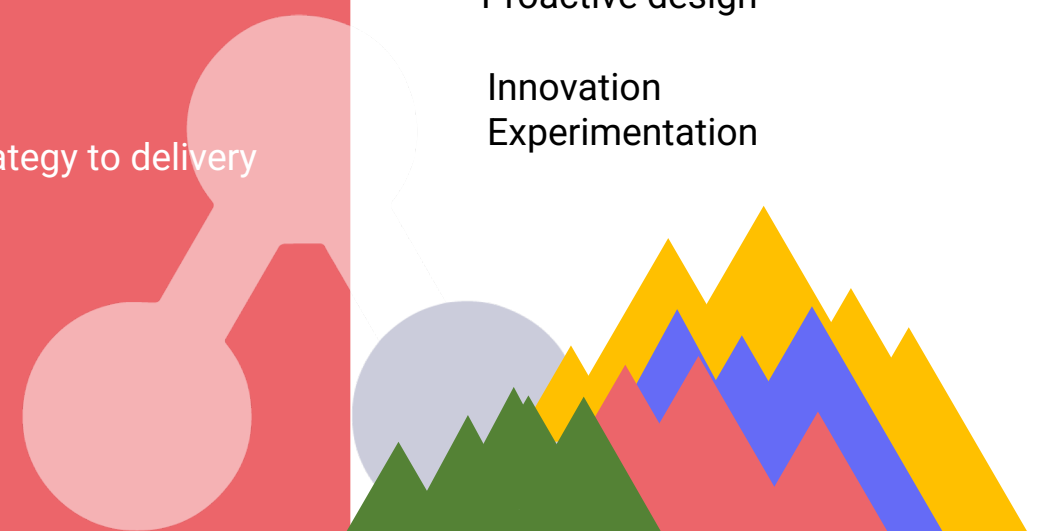
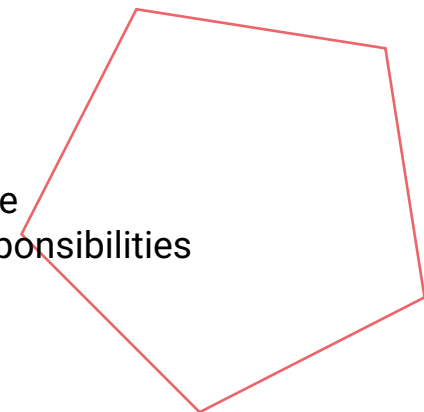
Domain governance

Goals, OKRs, KPIs
From business strategy to delivery

Agile reorganisation
Evolve solution architecture
Layered architecture & responsibilities
Agile architecture

Facilitate change
Proactive design

Innovation
Experimentation





How we did it

Introduced tool to monitor architectural drift and debt

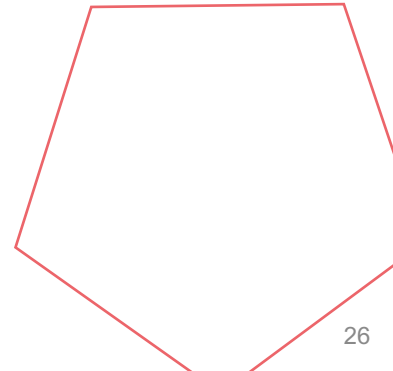
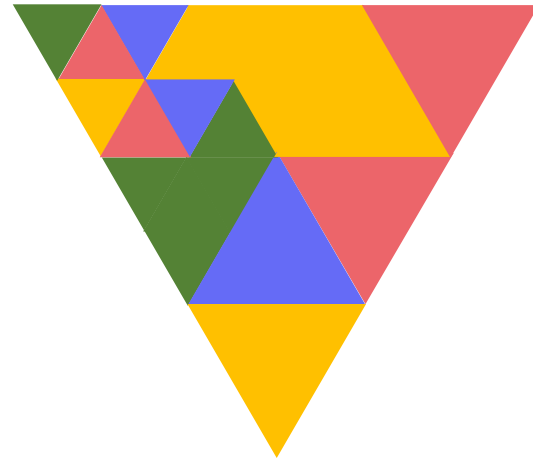
Enabled teams to view and understand their architecture state

Strengthened solution architecture as a discipline and artifact that team should actively take care of

Reduced need for re-factoring

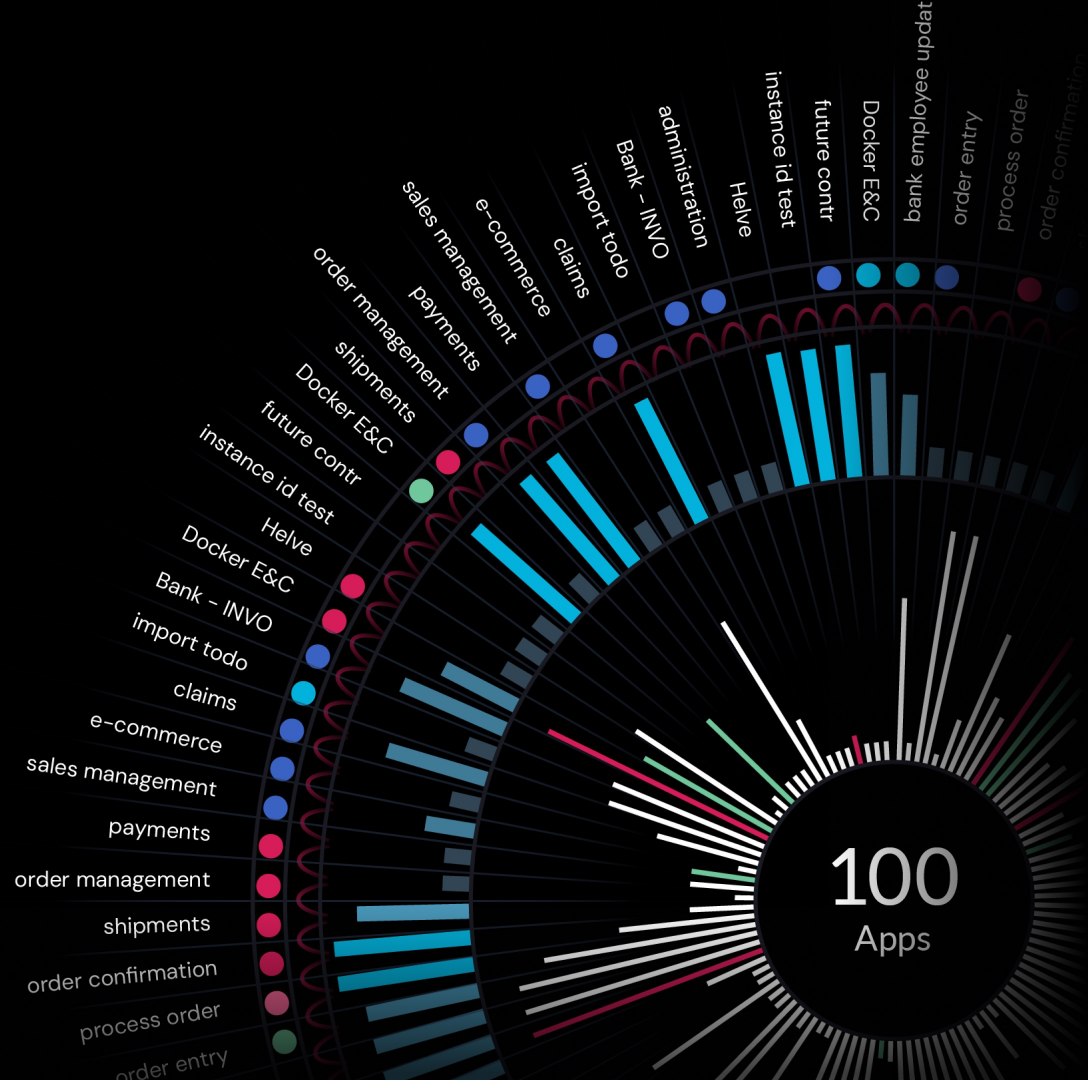
Increased quality of deployments and end product

Increased stability and resilience





Architectural Observability



Some Controversial Statements



Microservices is not a warranty for good architecture

You can have amazing code, with horrible architecture

Strangler-fig is (maybe) a solution for loosely coupled services

Some Controversial
Statements



Microservices is ~~not~~ a warranty for ~~good~~ architecture

bad

You can have amazing code, with horrible architecture

Strangler-fig is (maybe) a solution for loosely coupled services

Some Controversial
Statements



Microservices is ~~not~~ a warranty for ~~good~~ architecture

bad

software

You can have amazing code, with horrible ~~architecture~~

Strangler-fig is (maybe) a solution for loosely coupled services

Some Controversial Statements



Microservices is ~~not~~ a warranty for ~~good~~ architecture
software

bad

You can have amazing code, with horrible ~~architecture~~

GenAI will improve your code, not your software

Strangler-fig is (maybe) a solution for loosely coupled services

Some Controversial Statements



Microservices is ~~not~~ a warranty for ~~good~~ architecture
software

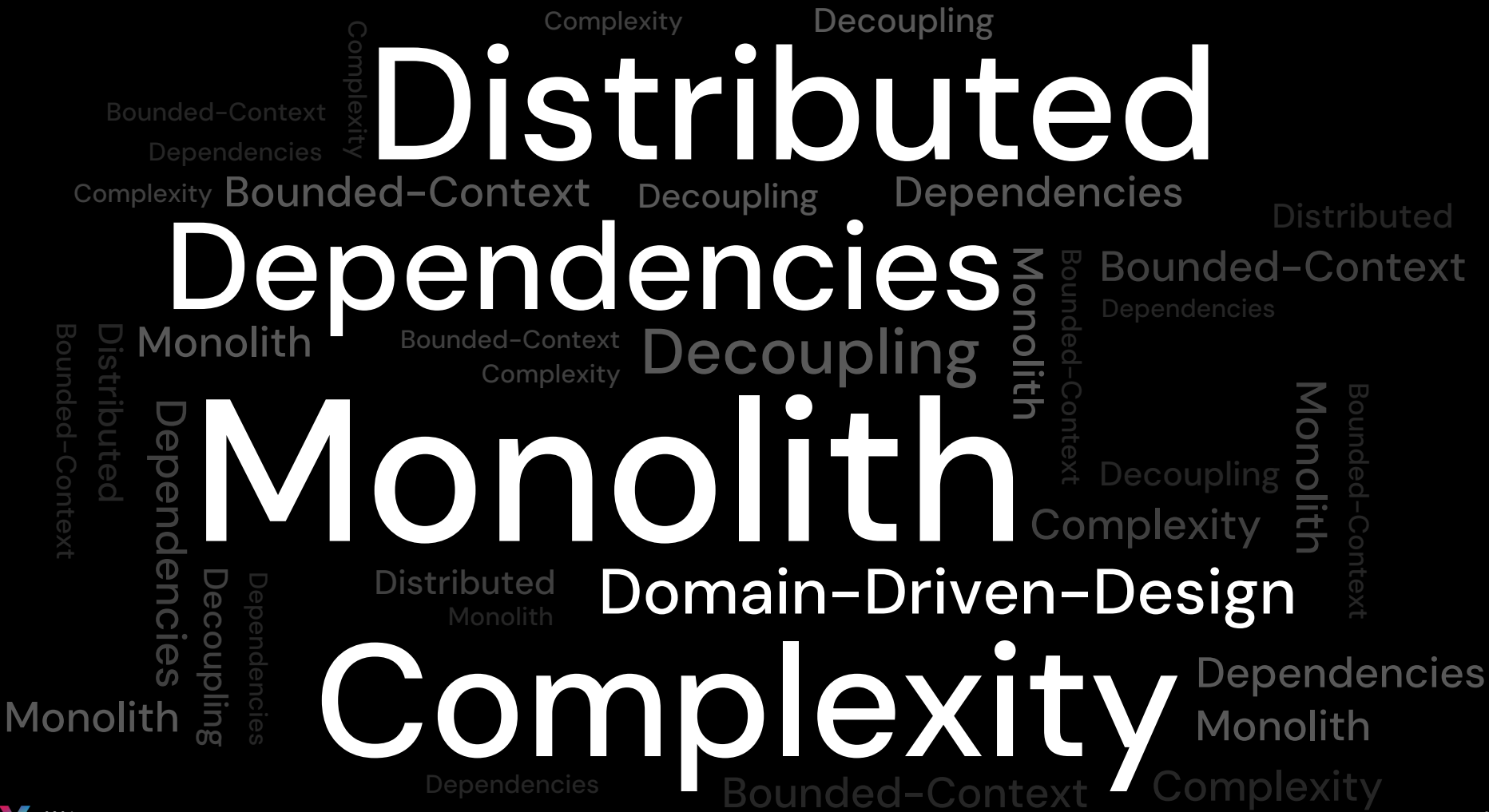
bad

You can have amazing code, with horrible ~~architecture~~

GenAI will improve your code, not your software

Strangler-fig is ~~(maybe)~~ a solution for loosely coupled services

not even



“By 2026, 80% of technical debt will be architectural technical debt”

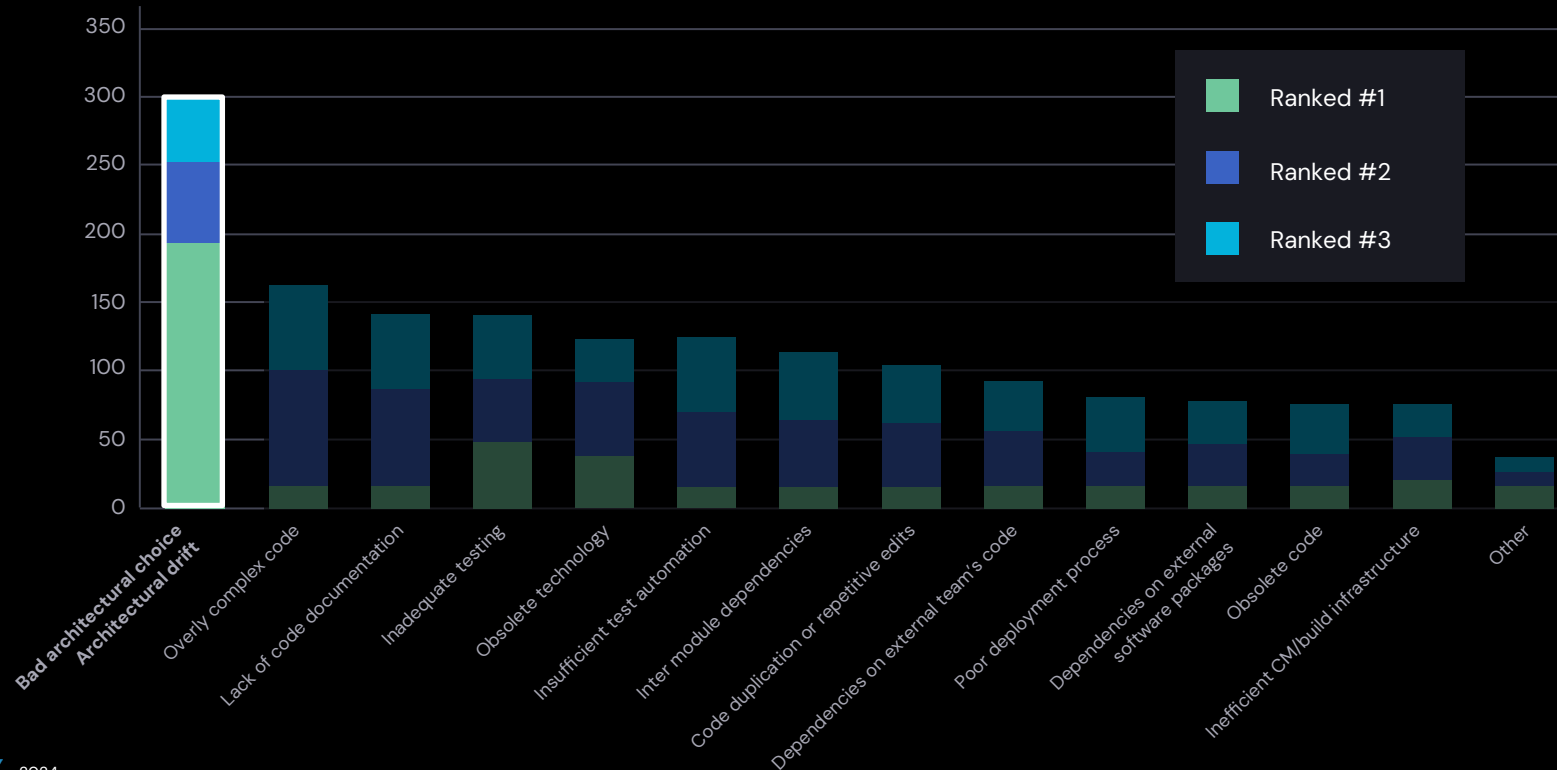
Source: Gartner, Measure and Monitor Technical Debt With 5 Types of Tools, Tigran Egiazarov, Thomas Murphy, 27, February 2023

“To help their organizations to successfully measure and monitor technical debt, software engineering leaders should:

Prevent time-consuming architectural rework by introducing tools to analyze architectural technical debt and monitor the amount of debt in their software architecture”

Ranking sources of technical debt

Fortune 500 & U.S. government survey of 1,800 professionals



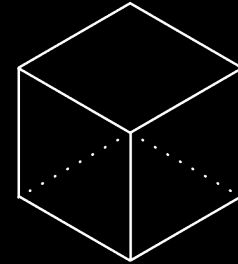
Source: Carnegie Mellon University Software Engineering Institute. Neil A. Ernst, Stephany Bellomo, Ipek Ozkaya, Robert L. Nord, and Ian Gorton, 8/2015

Whether monolithic or distributed, architecture is important.

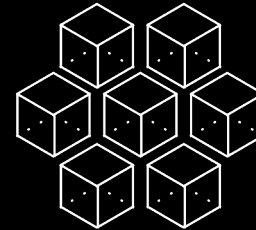
Organizations need to:

- **Understand** the architecture of applications
- **Track** architectural drift
- **Refactor** iteratively and improve quality

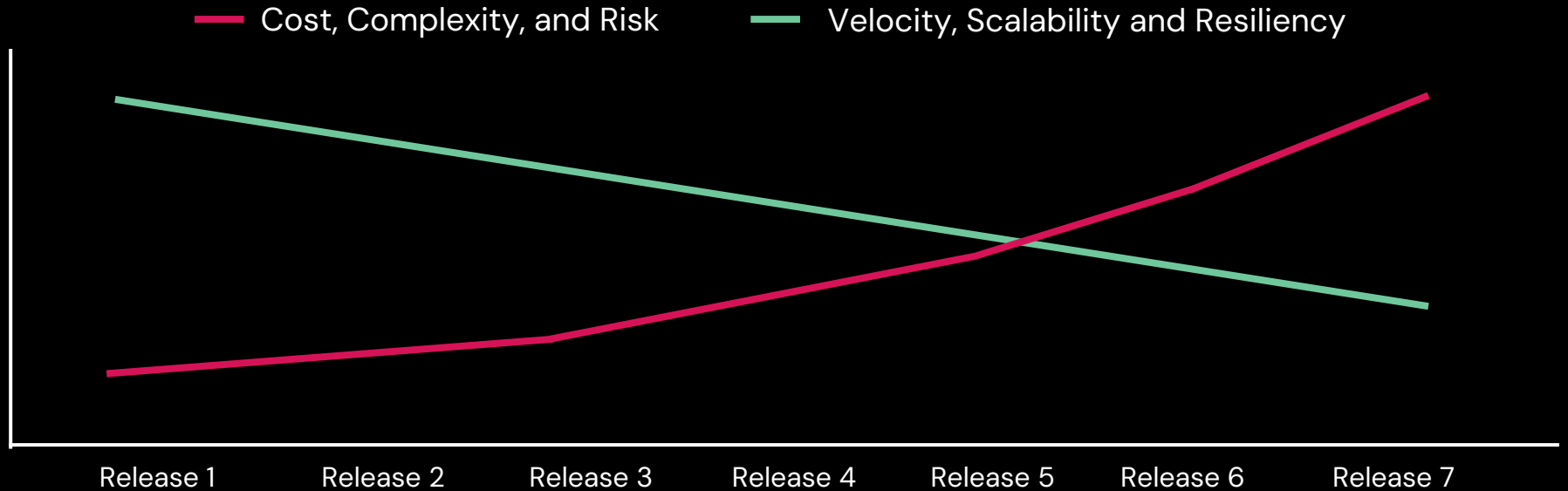
Monoliths



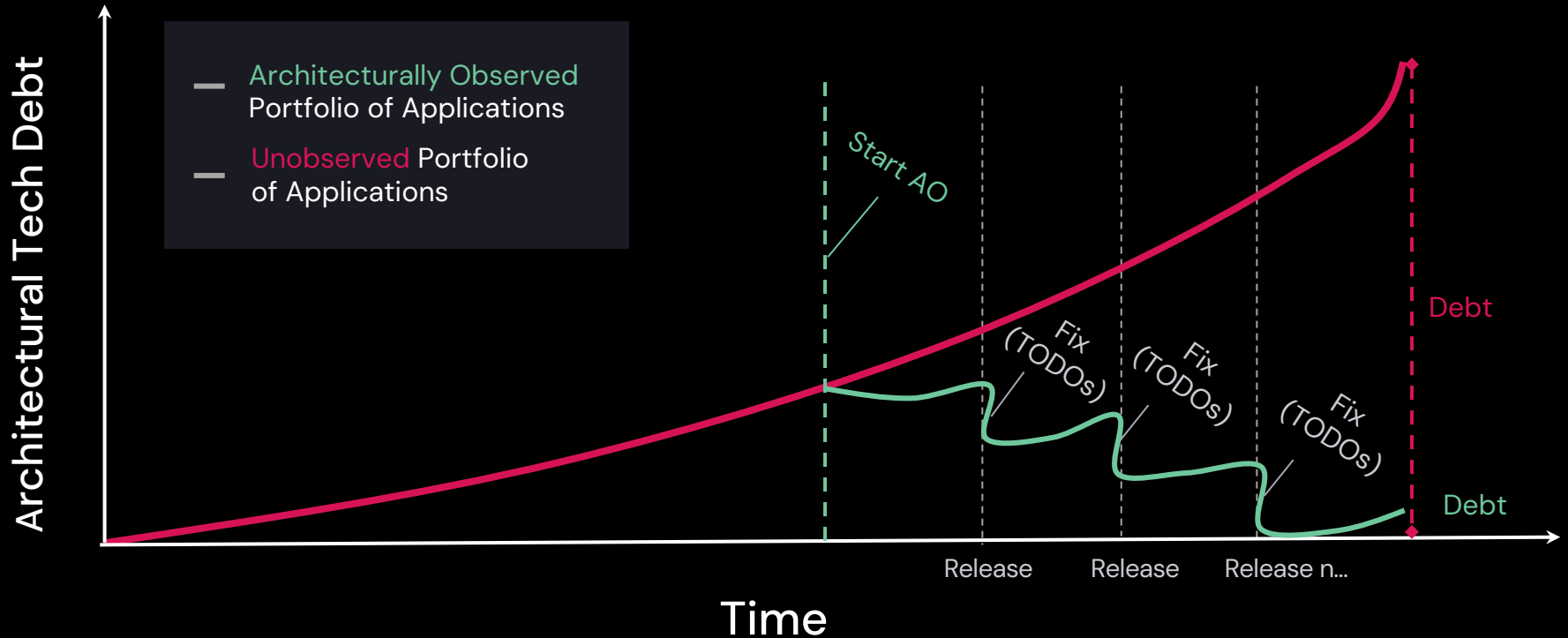
Microservices



Without dealing with architecture, applications need modernizing every five years



The architectural observability concept





Architectural debt is minimized
by having less dependent elements

Microchain architecture is about clearing
dependencies

Types of dependencies



Runtime

API calls,
domain calls, method
calls, DB tables



Compile-time/ deploy-time

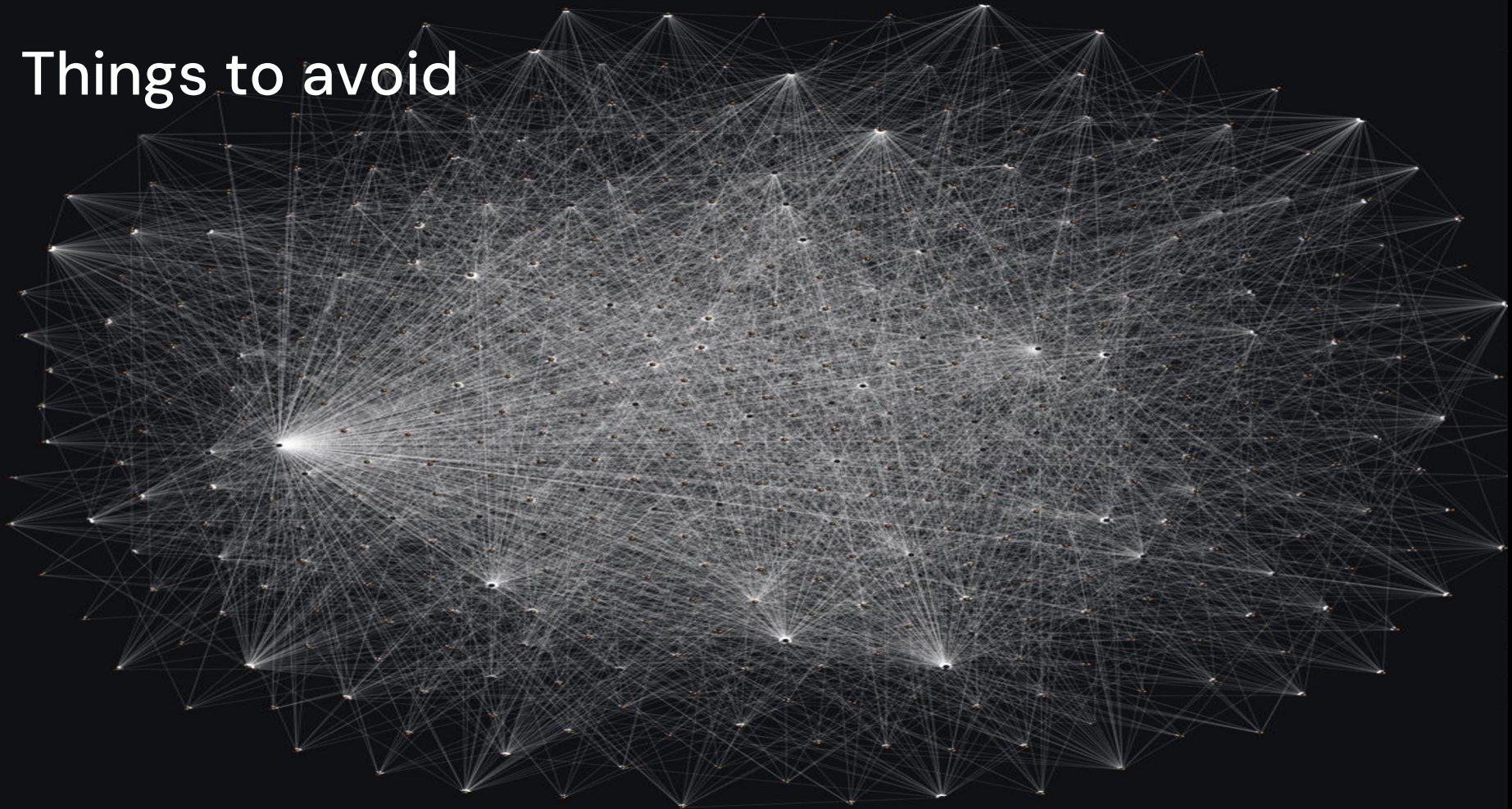
Classes, libraries,
binaries



Unneeded dependencies

Dead code,
unused libraries

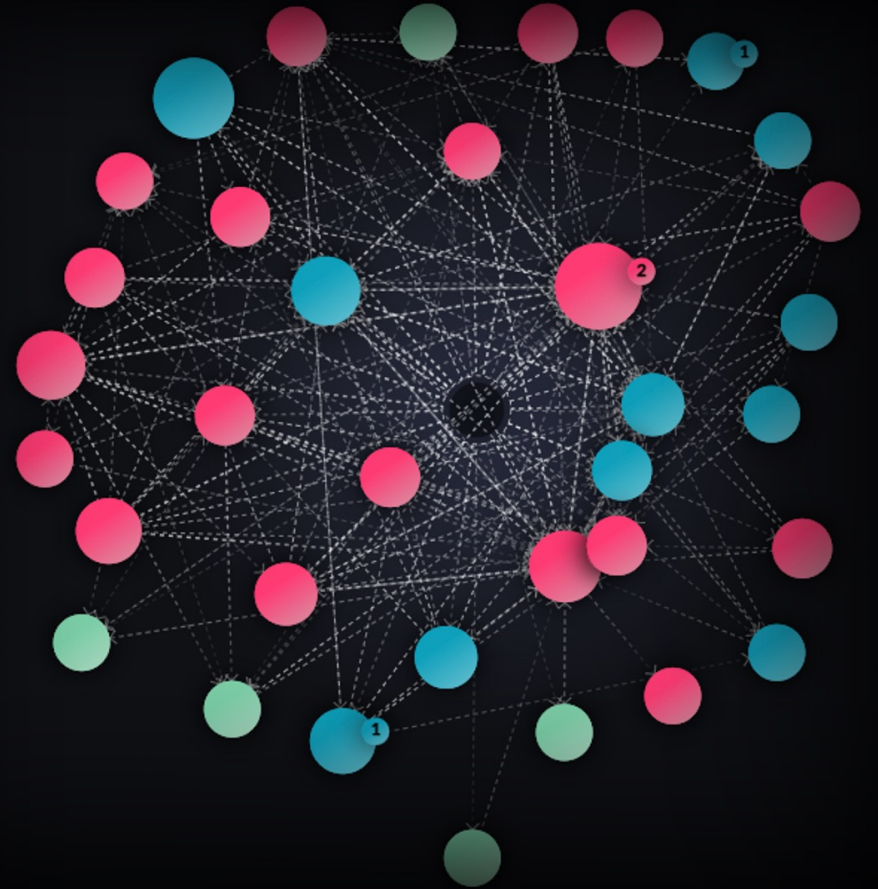
Things to avoid



Things to avoid

~~Complete mesh~~

Complete mess



Key Takeaways

1. Architectural technical debt is important
2. Observe architecture in your SDLC
3. Iteratively refactor and avoid "modernization"



When it's hot,
It's great to be Cool.

Visit us at
booth 211

Join us for beer and wine



Learn more
vfunction.com

The GARTNER COOL VENDOR badge is a trademark and service mark of Gartner, Inc., and/or its affiliates, and is used herein with permission. All rights reserved. Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings or other designation. Gartner research publications consist of the opinions of Gartner's Research & Advisory organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.