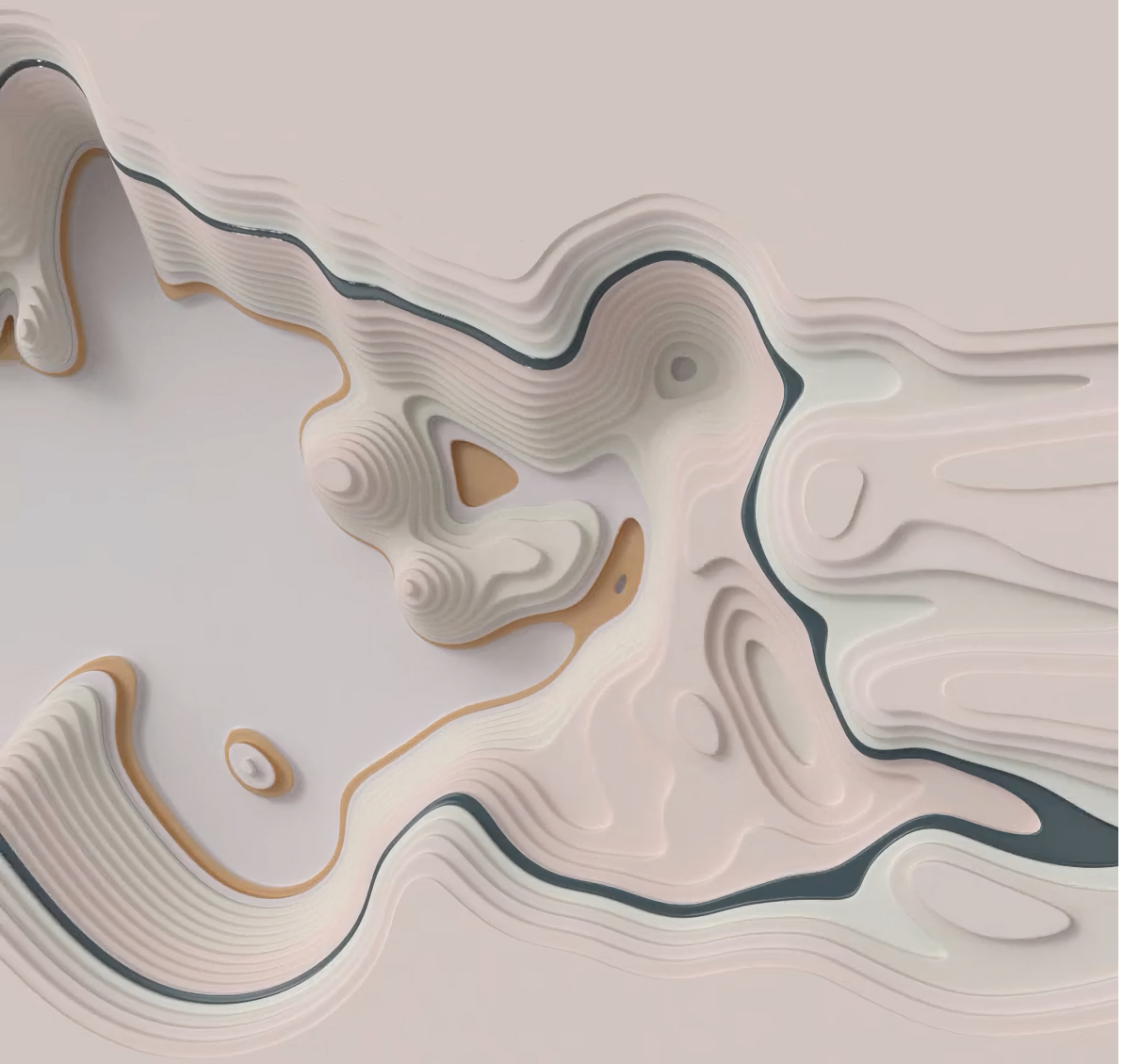


Software Architecture Matters

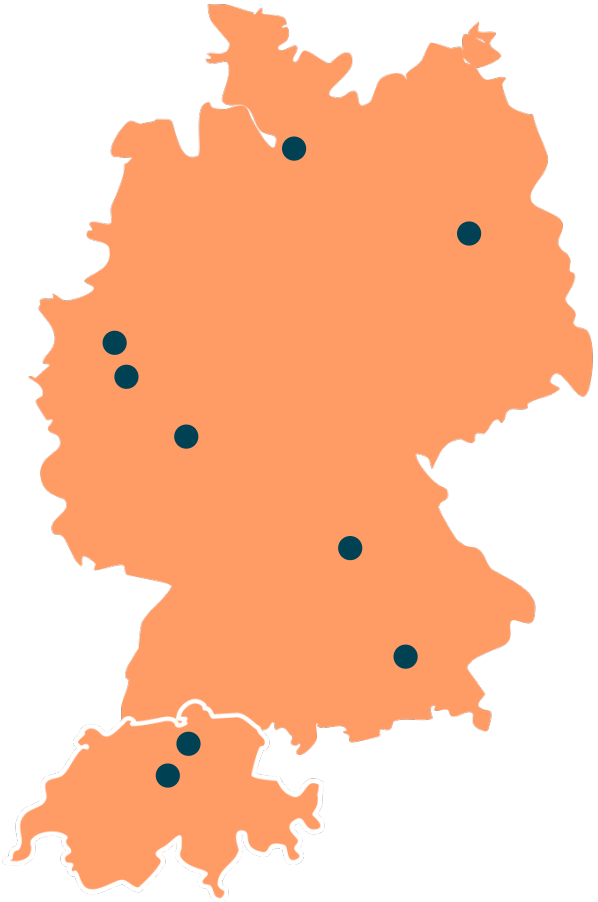
Stefan Tilkov
stefan.tilkov@innoq.com
@stilkov{@innoq.social}

INNOQ

Company Background

INNOQ

Founded	Employees	Customers	Annual sales 2022
1999	175+	300+	~ € 20 M



SECTORS Finance | TC | Logistic | E-commerce | Fortune 500 | SME | Start-ups

AXA SACAC ebay VORWERK breuninger mobile.de

What is “Software Architecture”?

Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

ISO 42010

“Stop putting me to sleep with technology”

Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change

Grady Booch

Time to market
Usability and Performance
Security and Availability
Privacy and GDPR
Organizational Agility

If you consider technology relevant, you have to invest in architecture

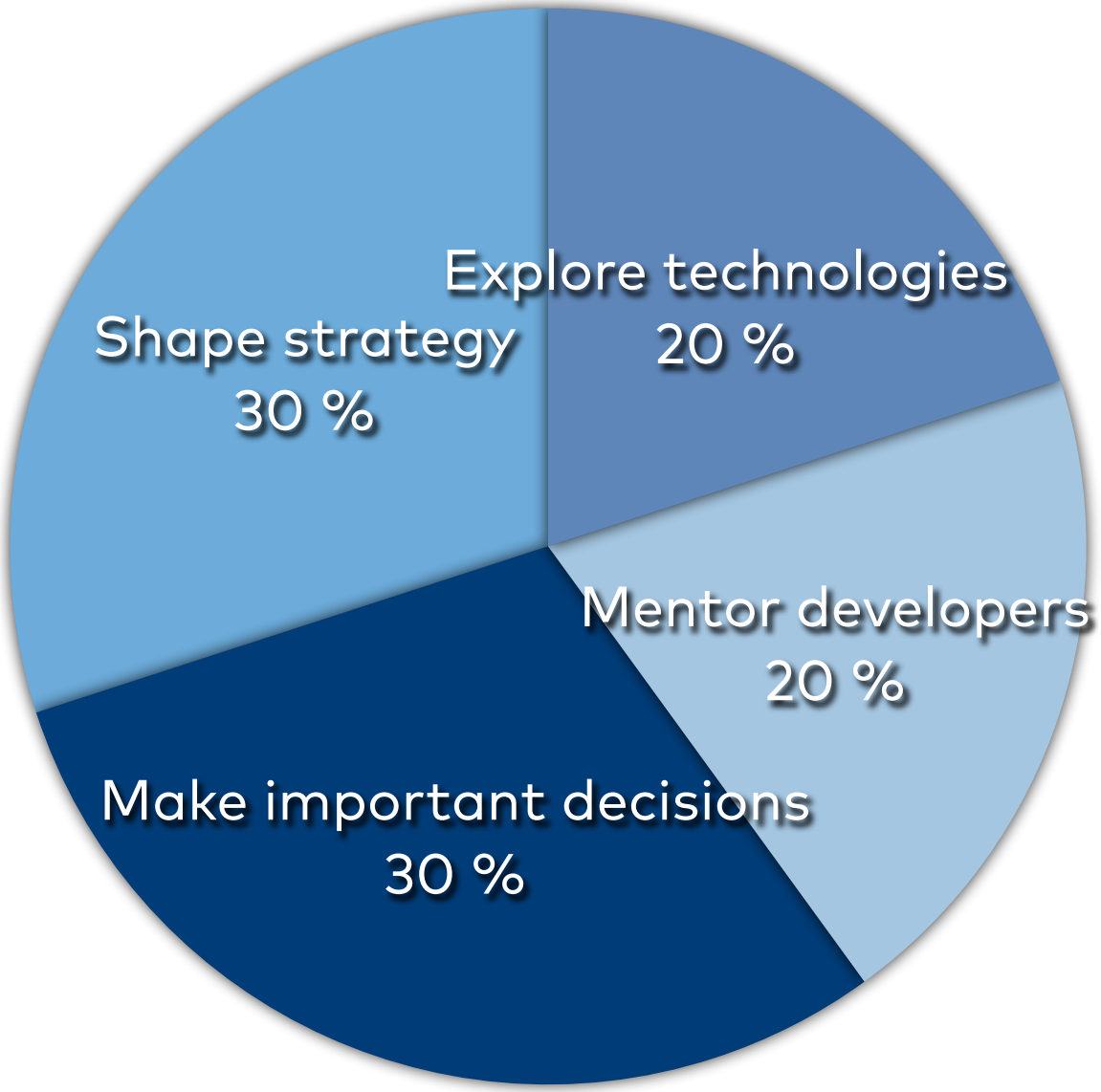
Your architecture, as the sum of the most critical decisions, will determine the fate of your system

Architecture is not an upfront activity performed by somebody in charge of telling everyone else what to do

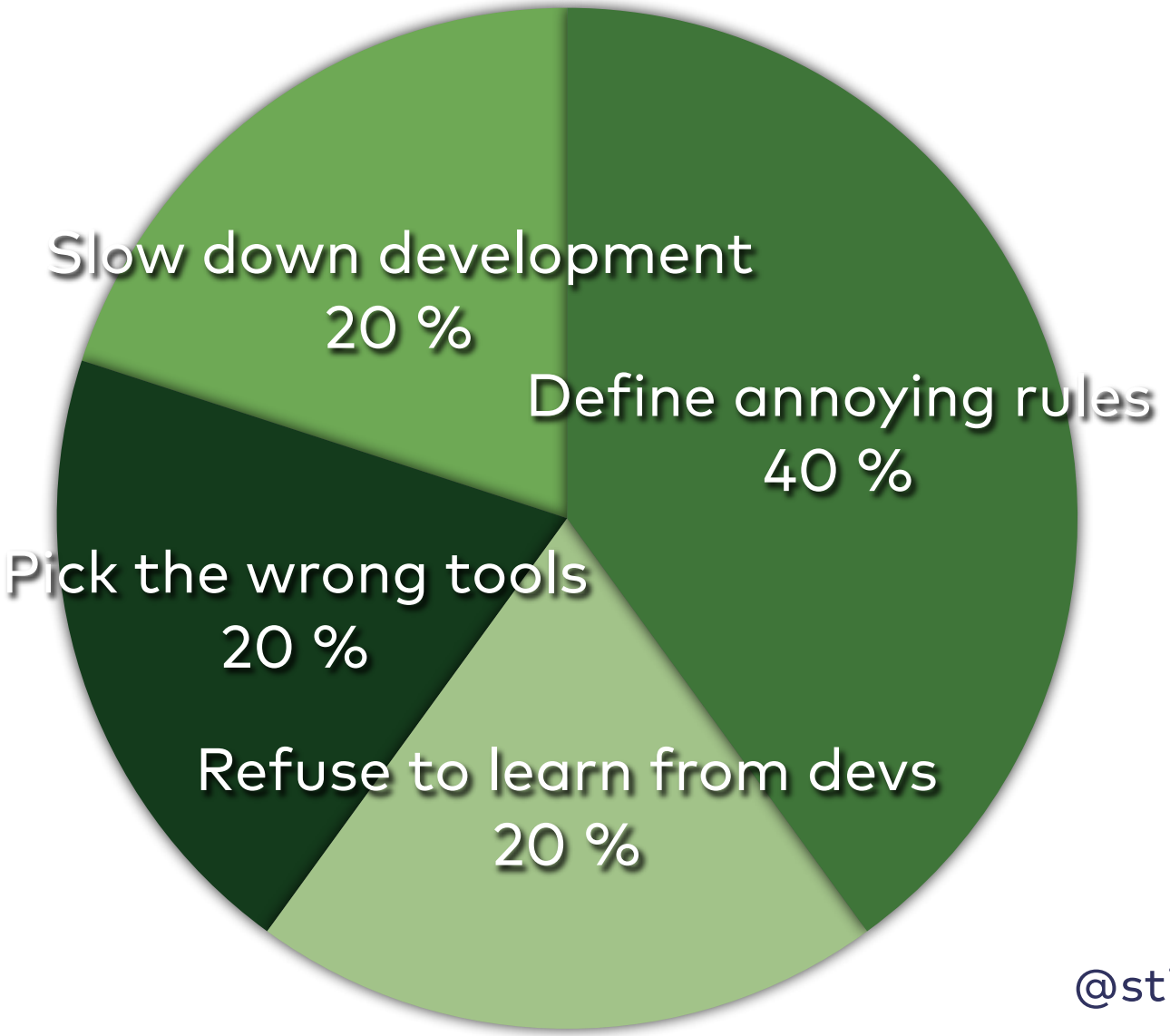
Architecture is a property of a system, not a description of its intended design

How do architects work?

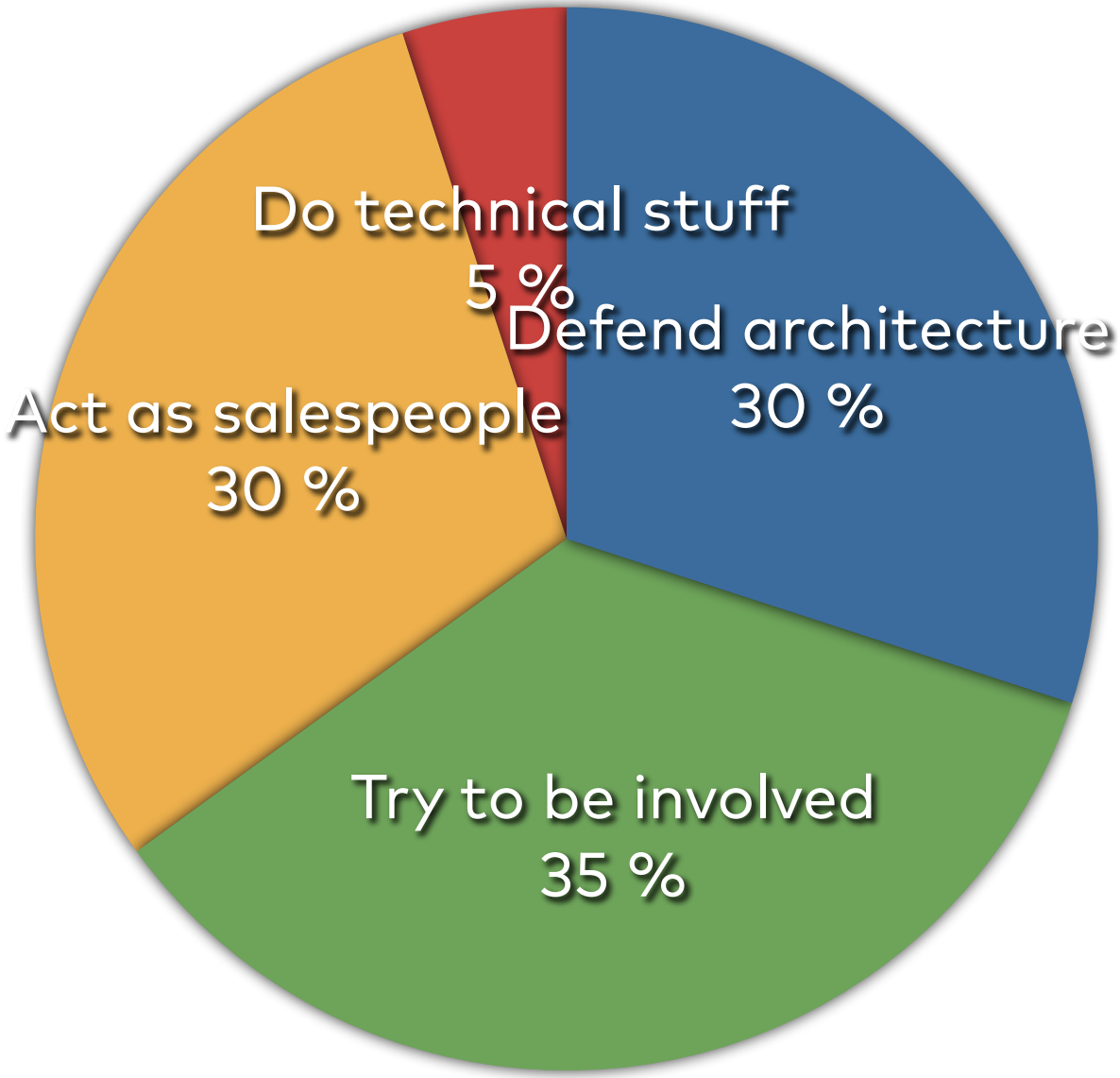
What architects want to do



What others think architects do



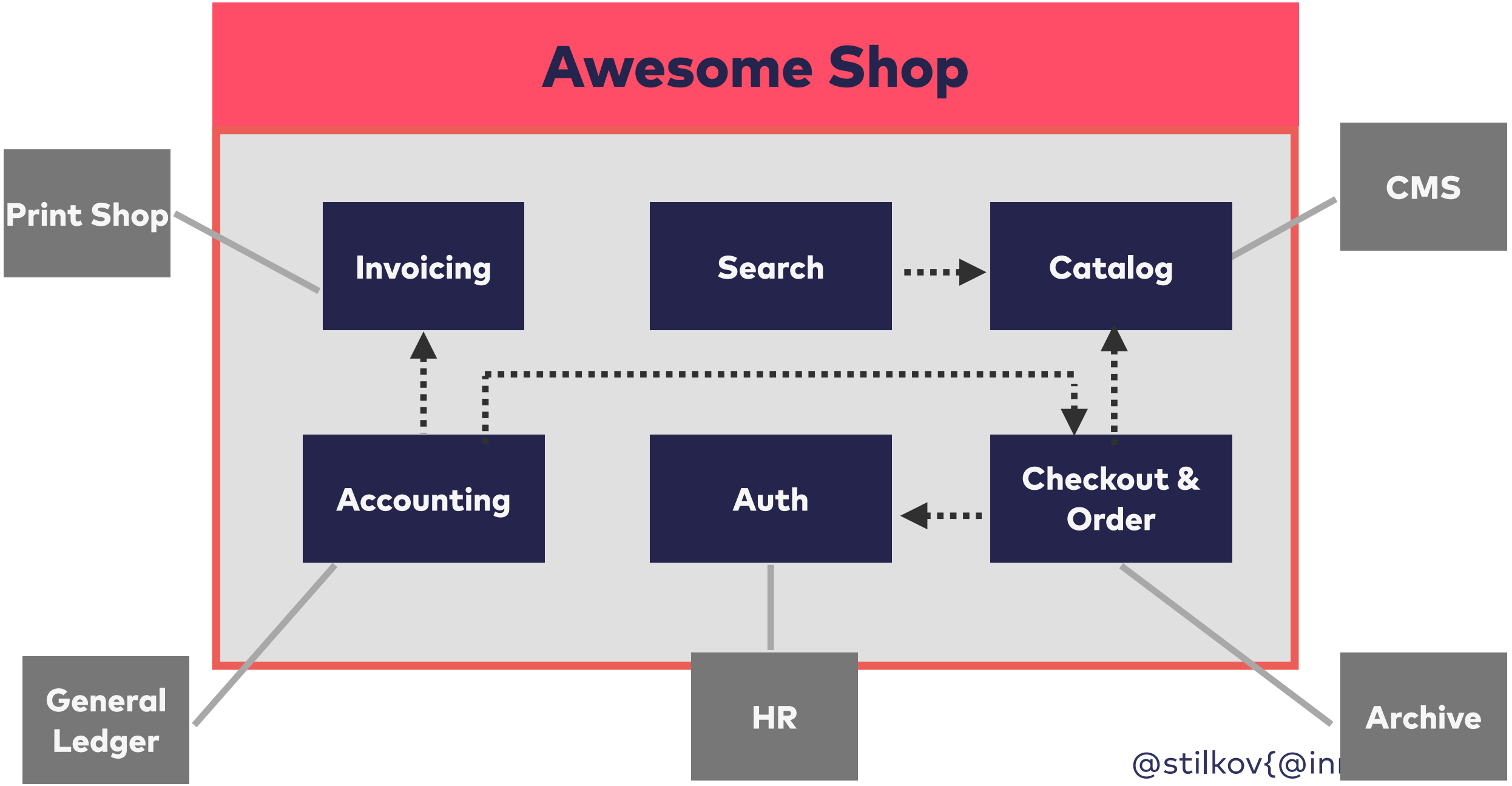
What good architects actually do

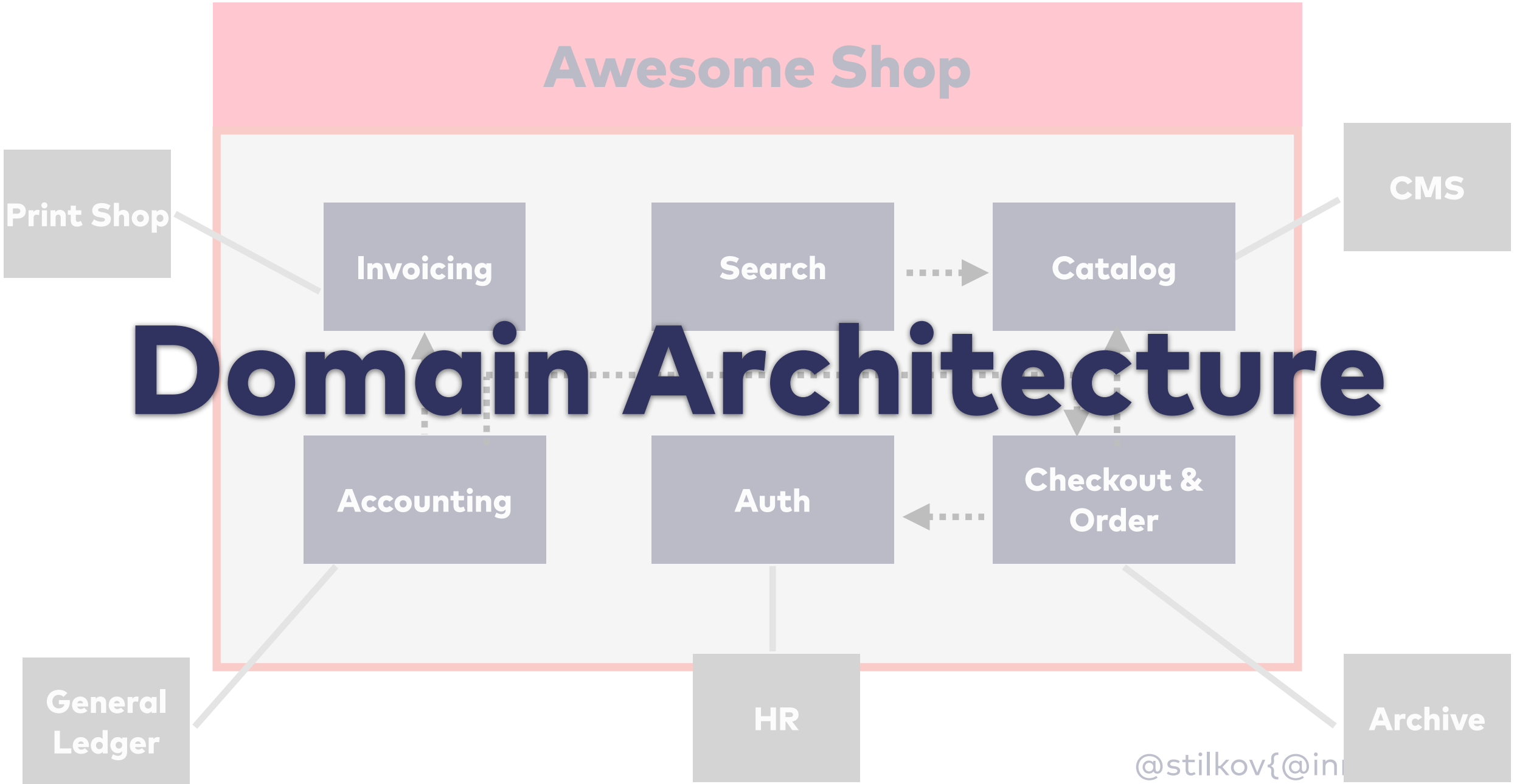


7 Recommendations for successfully architecting (not only) in the large

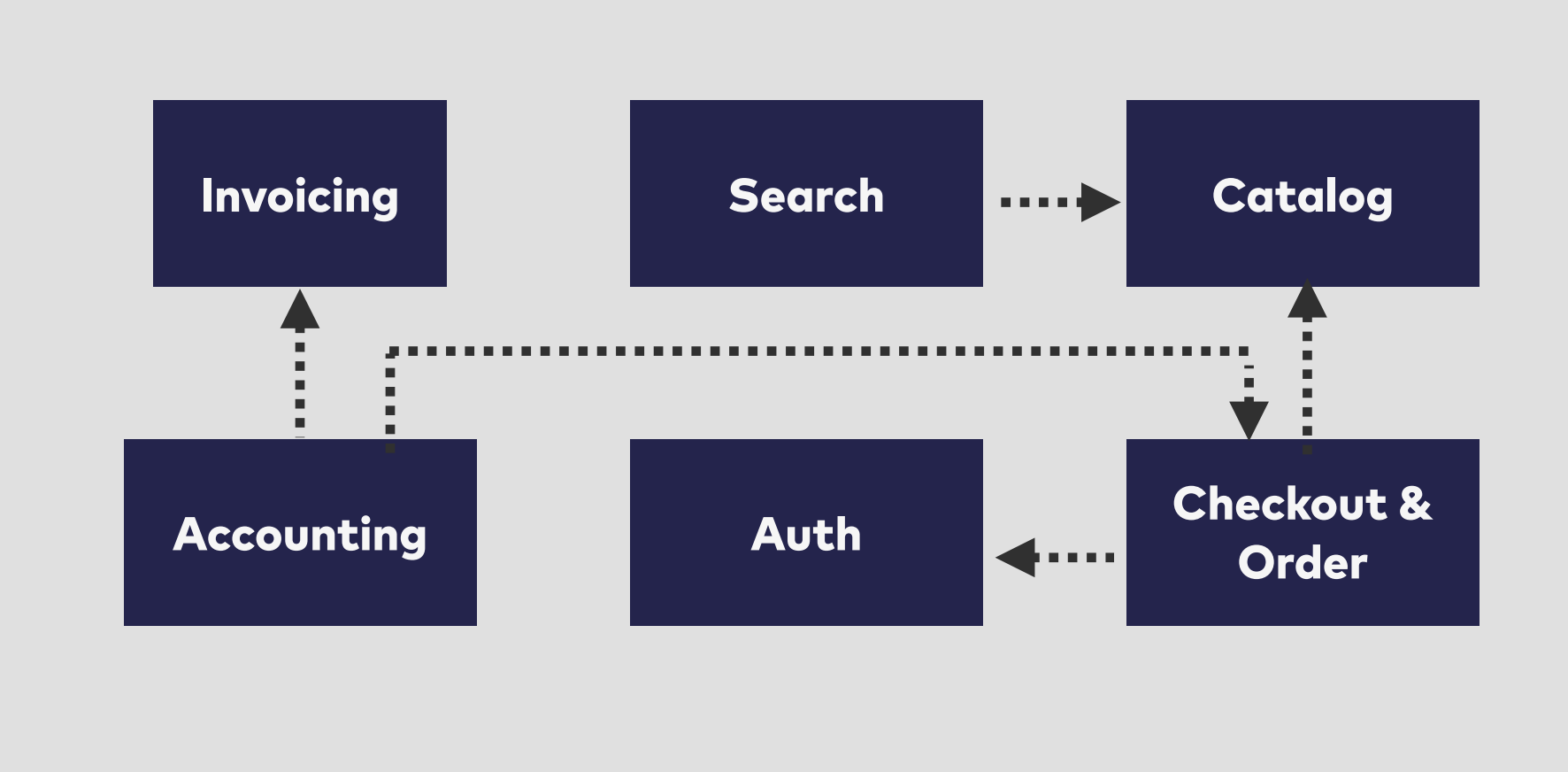
1.
Choose your perspective(s) consciously

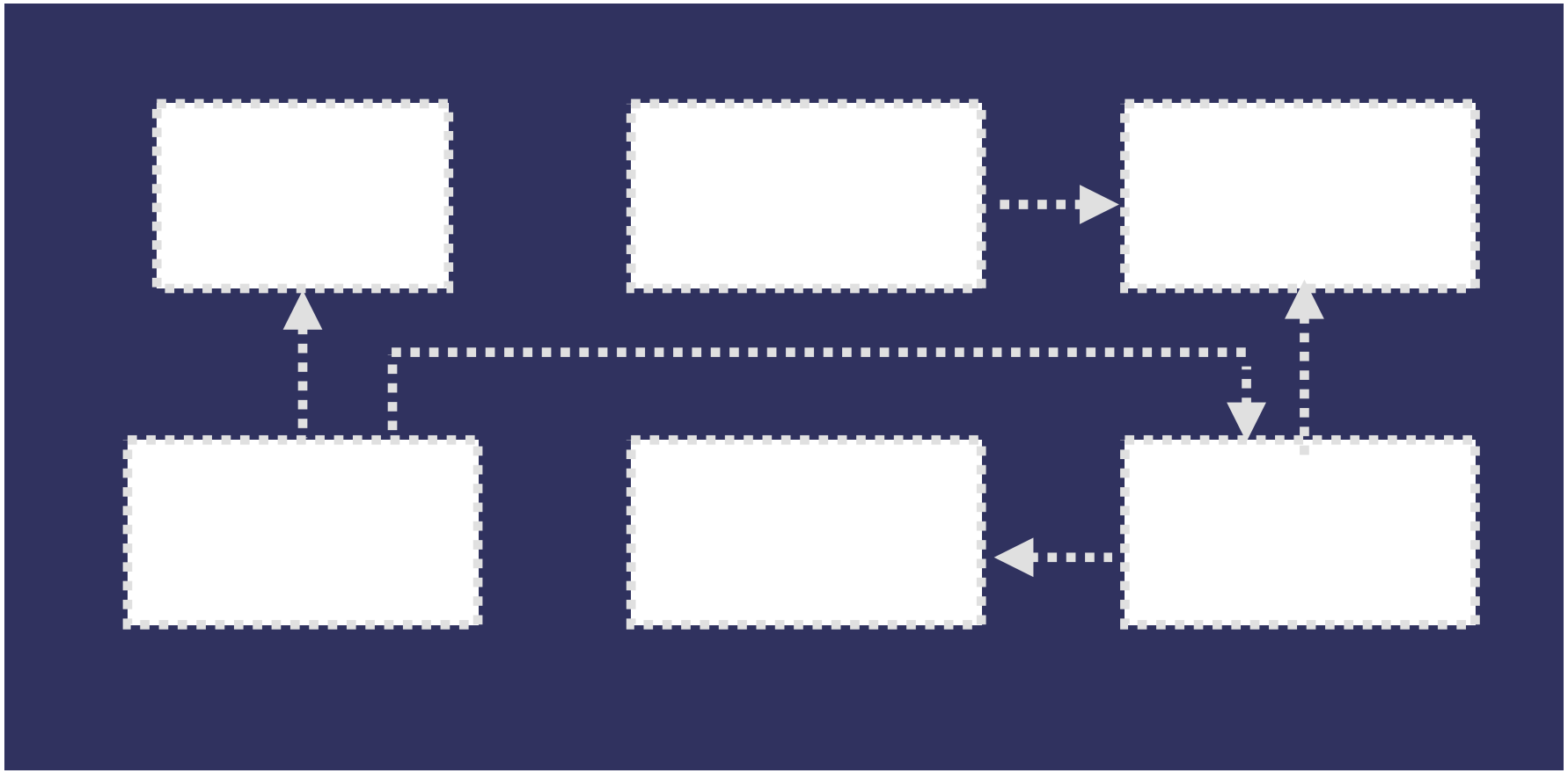

Print Shop
CMS
Awesome Shop
General Ledger
Archive
HR


Awesome Shop
Print Shop
Invoicing
Search
Catalog
CMS
Accounting
Auth
Checkout & Order
General Ledger
HR
Archive



Domain Architecture


Invoicing
Search
Catalog
Accounting
Auth
Checkout &
Order



Macro Architecture

Ruby on Rails
MySQL

NodeJS
ElasticSearch

COTS

Java
Spring Boot

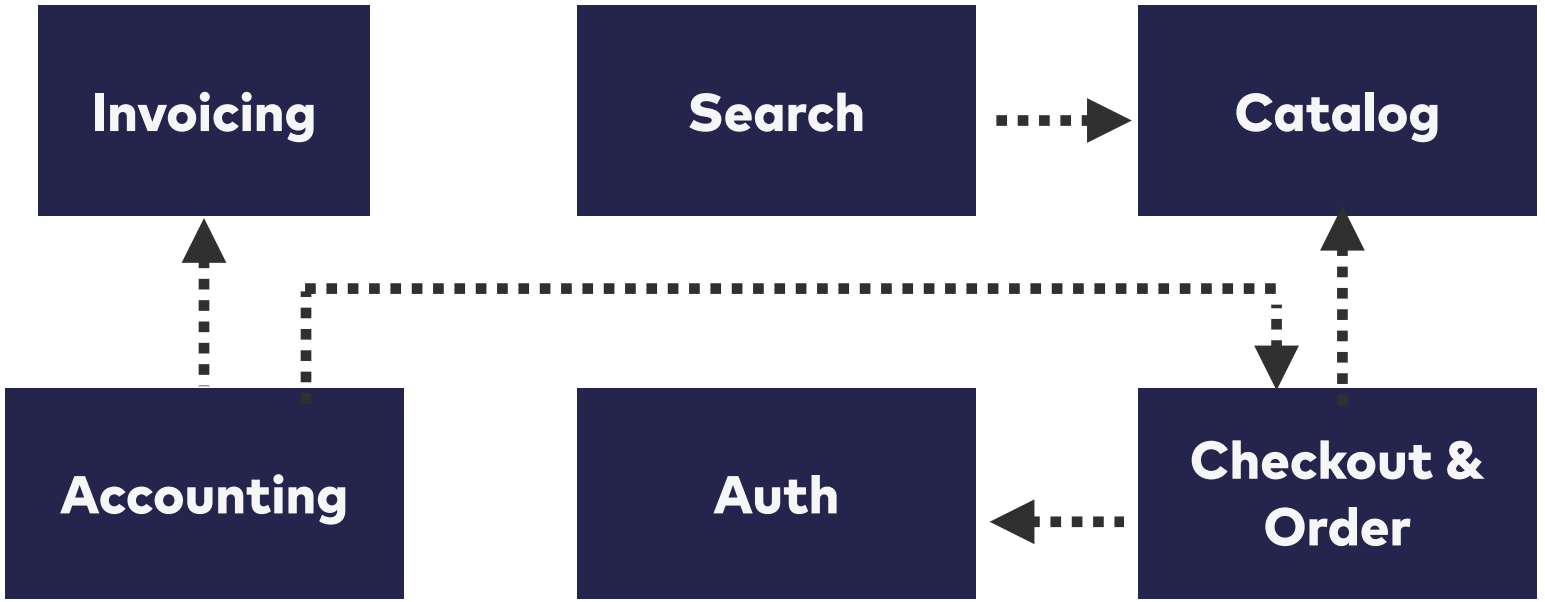
OSS Product

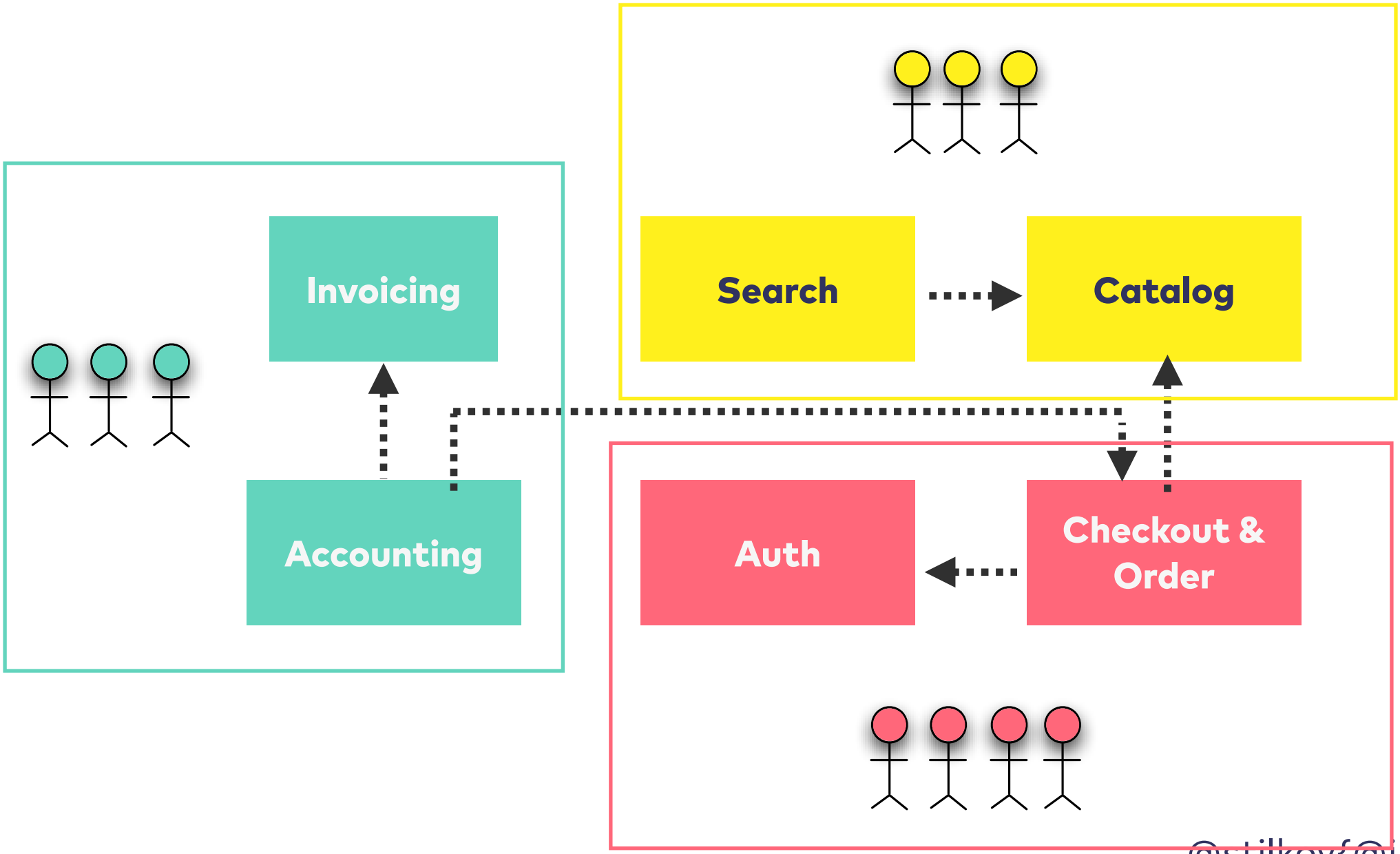
Java
Spring Boot

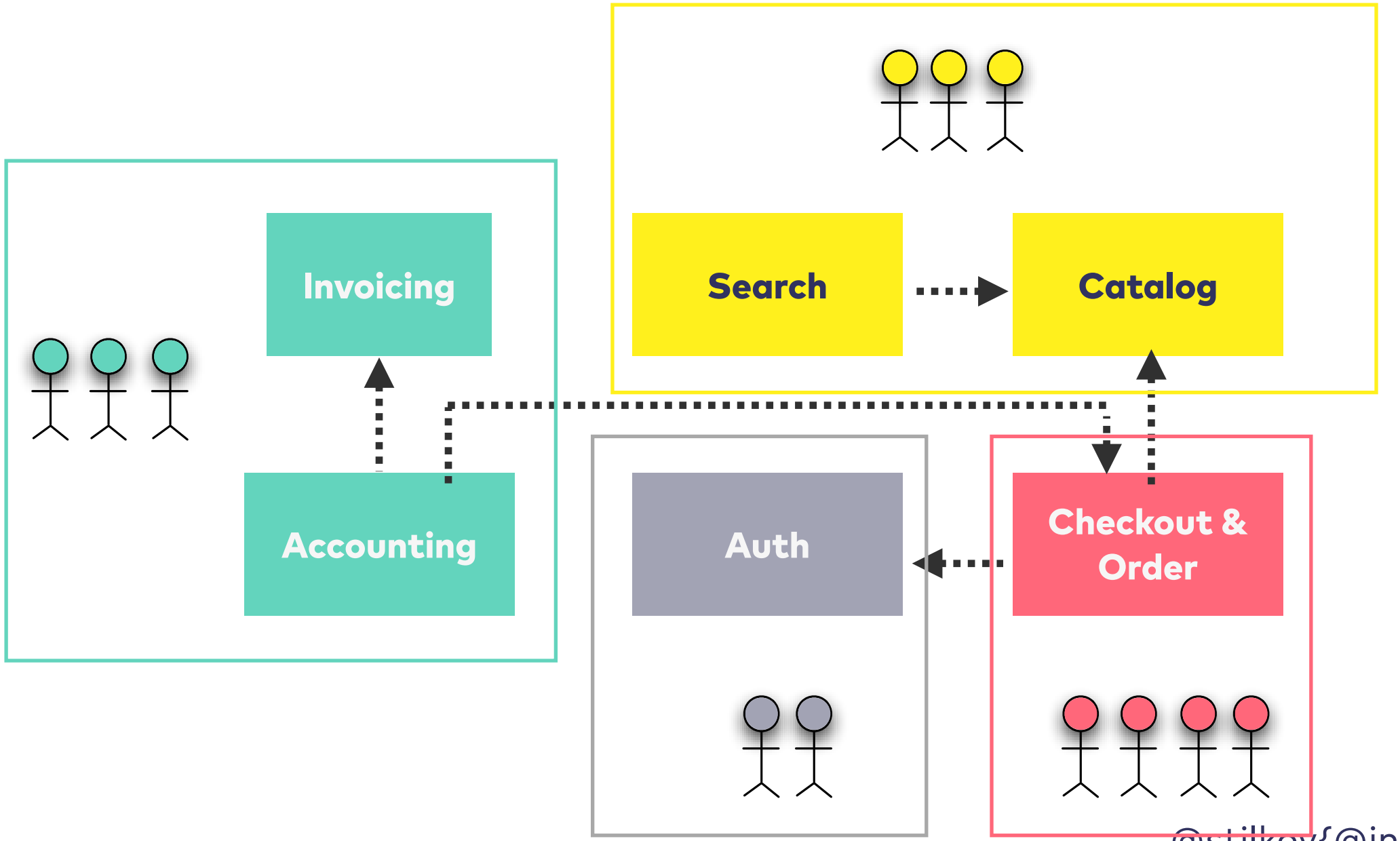

Ruby on Rails
MySQL
NodeJS
ElasticSearch
COTS
Micro Architecture
Java
Spring Boot
OSS Product
Java
Spring Boot

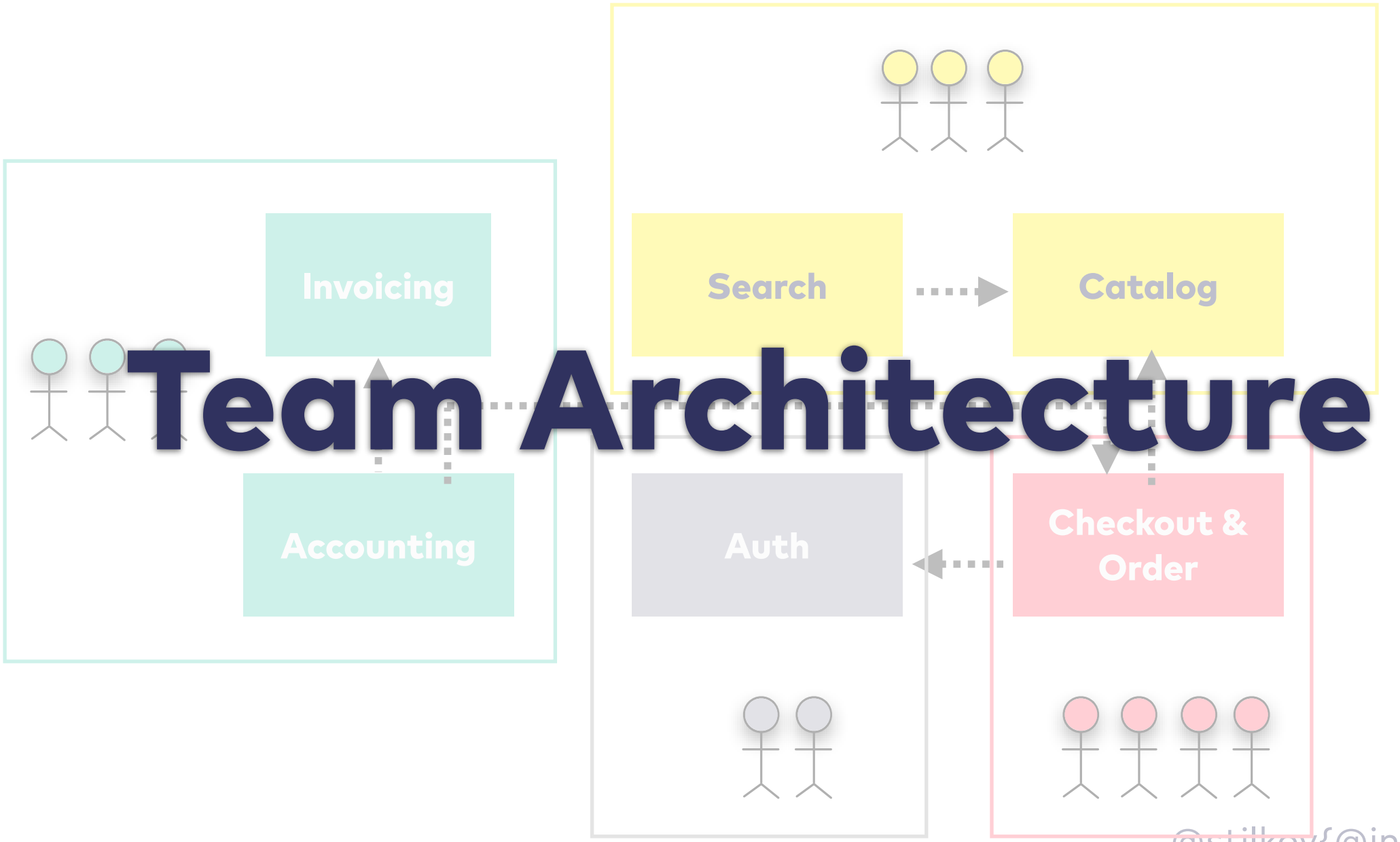
Be aware of each architecture perspective's focus

2.
Explicitly architect your team setup


Invoicing
Search
Catalog
Accounting
Auth
Checkout &
Order


Invoicing
Accounting
Search
Catalog
Auth
Checkout &
Order


Invoicing
Accounting
Search
Catalog
Auth
Checkout &
Order

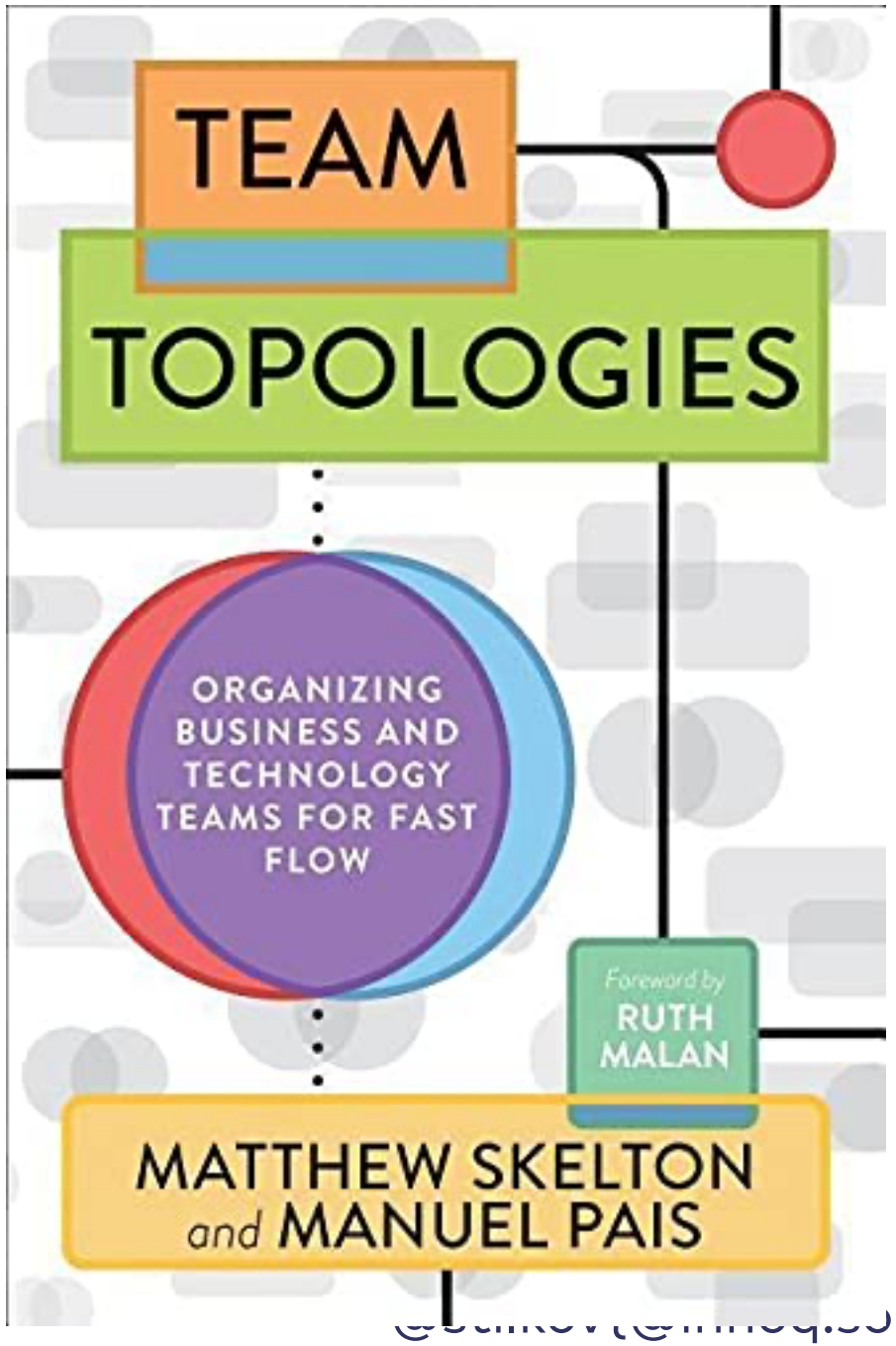


Team Architecture

About Team Topologies

Team Topologies is the leading approach to organizing business and technology teams for fast flow, providing a practical, step-by-step, adaptive model for organizational design and team interaction.

https://teamtopologies.com/



Fundamental topologies

- Stream-aligned team
- Enabling team
- Platform team
- Complicated-subsystem team

Interaction modes

- Collaboration
- X-as-a-Service
- Facilitating

Things that are great about team topologies

- Explicit team-first approach
- Autonomy at the heart of value creation
- Technology-agnostic
- Long-lived teams instead of project thinking
- Based on actual experience, research, collaboration

3.
Match your organizational setup to your project size, context, and culture

Single team

- Some tasks are architecture tasks
- No explicit architecture roles

Small number of collaborating teams

- Architecture board
- Teams send members

Large number of independent teams

- Explicit roles
- Separate arch team
- Arch team sends members to teams

4.
Don't be afraid to decide things centrally

Things that need to be decided centrally

- What to centralize and what to leave to individual teams
- Which teams exist and what their responsibilities are
- Anything relevant* at the seams of more than one team
- Necessary* global policies and strategic* aspects

*for appropriate values of relevant, necessary, and strategic

5.
Pick your battles wisely

Don't do everything at once

- Change programming language and development environment
- Switch from RBDMS to alternative models
- Move to the cloud
- Replace desktop UIs with web frontends
- Introduce asynchronous messaging
- Switch to microservices or whatever this year's hype might be

“If you love somebody
set them free”

Sting

Be careful with standardization

	Standardized	Team decision
Programming Language	x	
Implementation frameworks & libraries		x
IDE		x
API standard	x	
Frontend stack		x
UI integration approach	x	
Operating environment	x	

Be careful with standardization

	Standardized	Team decision
Programming Language		x
Implementation frameworks & libraries		x
IDE		x
API standard	x	
Frontend stack	x	
UI integration approach	x	
Operating environment		x

Be careful with standardization

	Standardized	Team decision	Best practice
Programming Language		x	x
Implementation frameworks & libraries		x	x
IDE		x	x
API standard	x		
Frontend stack		x	x
UI integration approach		x	x
Operating environment	x		

6.
Enforce the least viable amount of rules, rigidly

Things you might want to enforce

- Interoperability
- Integration
- Operability
- Data analysis
- …

- Privacy
- Accessibility
- Security
- Compliance
- …

7.
Balance prescriptive vs. descriptive architecture

A well-documented architecture, with wonderful, up-to-date descriptions, explicit ADRs, beautiful diagrams, consistent models, created with state-of-the-art tools, can still be bad.

Good decisions will not necessarily satisfy everyone; most decisions that do are likely bad.

Compromising on essential aspects will likely avoid conflicts now, but hurt really bad later

Conclusion

Architecture remains the most interesting and fascinating area in IT because it matters. If you get it wrong, everything breaks down – yet even if you get it right, success is not guaranteed

Whatever you do, consider the context and beware of one-size-fits-all solutions

Don't aim for perfection – aim for evolvability

That's all I have

Stefan Tilkov
stefan.tilkov@innoq.com
+49 170 471 2625
@stilkov – @stilkov@innoq.social

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim
+49 2173 3366-0

Ohlauer Str. 43
10999 Berlin

Ludwigstr. 180E
63067 Offenbach

Kreuzstr. 16
80331 München

Hermannstrasse 13
20095 Hamburg

Erftstr. 15-17
50672 Köln

Königstorgraben 11
90402 Nürnberg

innoQ Schweiz GmbH

Gewerbestr. 11
6330 Cham

Hardturmstr. 253
8005 Zürich