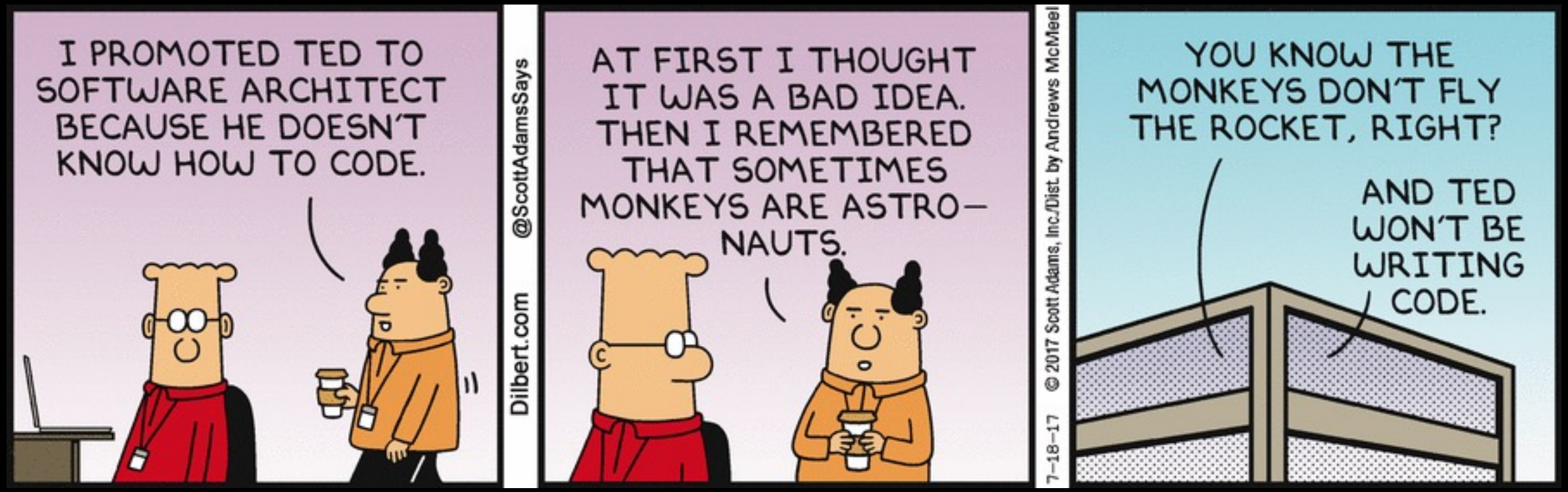

I PROMOTED TED TO SOFTWARE ARCHITECT BECAUSE HE DOESN'T KNOW HOW TO CODE.
Dilbert.com @ScottAdamsSays
AT FIRST I THOUGHT IT WAS A BAD IDEA. THEN I REMEMBERED THAT SOMETIMES MONKEYS ARE ASTRO-NAUTS.
7-18-17 © 2017 Scott Adams, Inc./Dist. by Andrews McMeel
YOU KNOW THE MONKEYS DON'T FLY THE ROCKET, RIGHT?
AND TED WON'T BE WRITING CODE.

Software architecture – Part 2

documentation

version: 1.0.3

Agenda

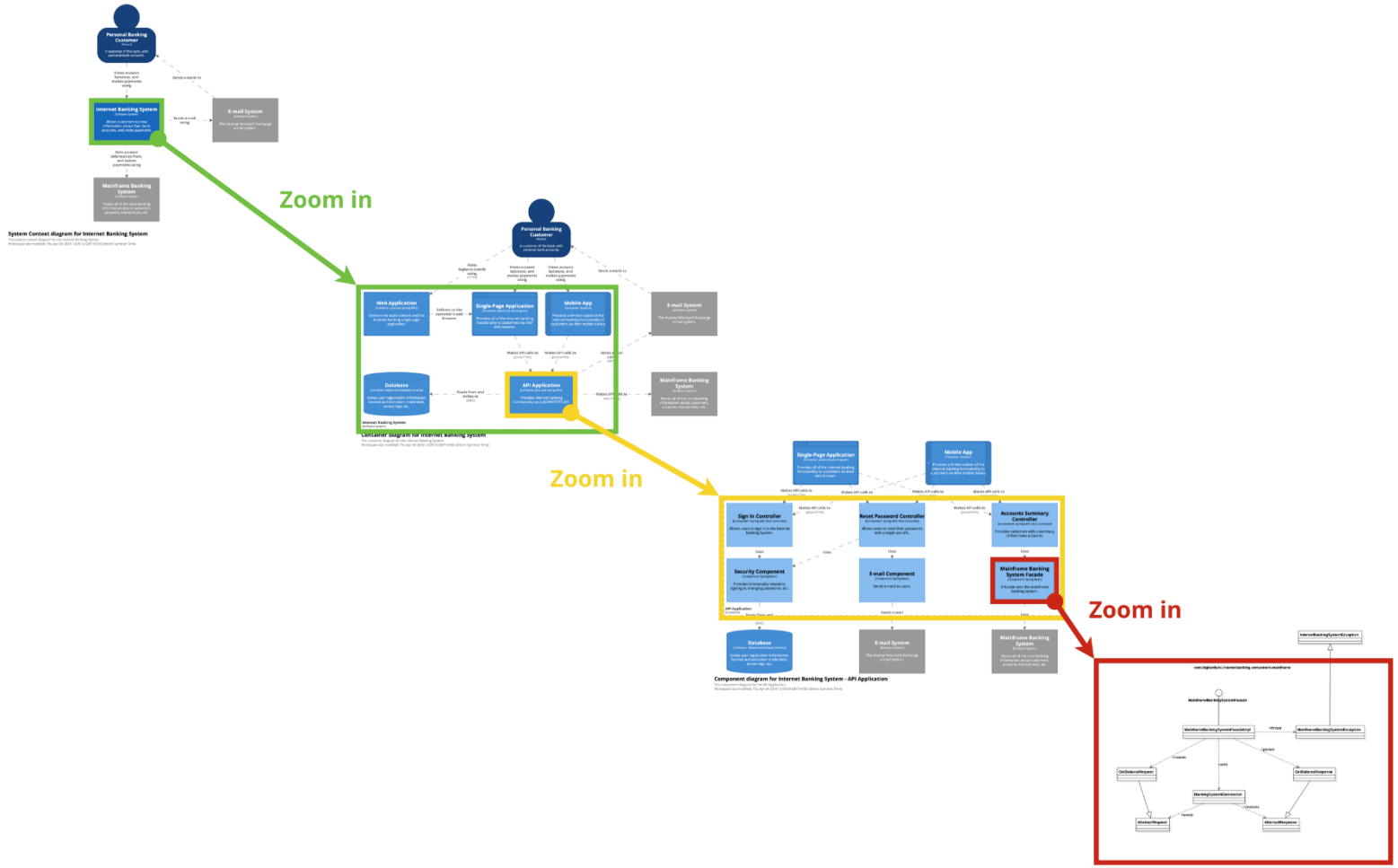
Software architecture documentation

- Recap Simon Browns C4 model
- N+1 Views
- UML diagrams

How to become a good architect

Recap: Simon Browns C4 model

C4 model by Simon Brown



Level 1
Context

Level 2
Containers

Level 3
Components

Level 4
Code

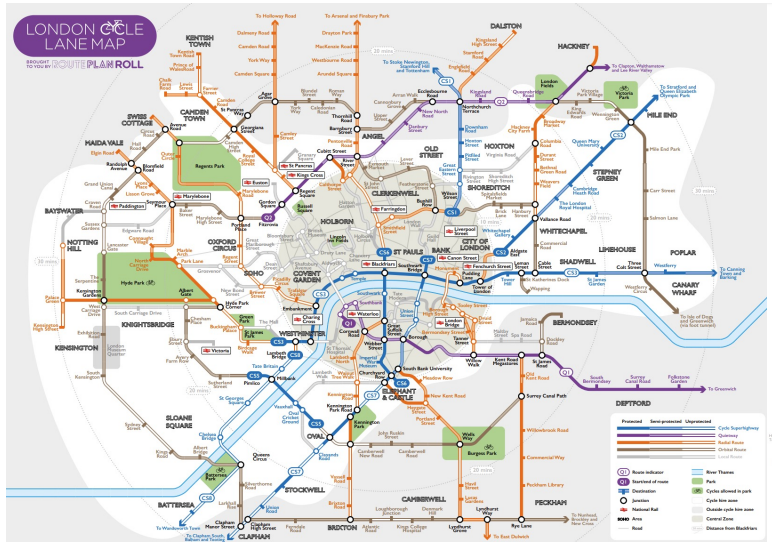
The C4 model is a way to document structure.

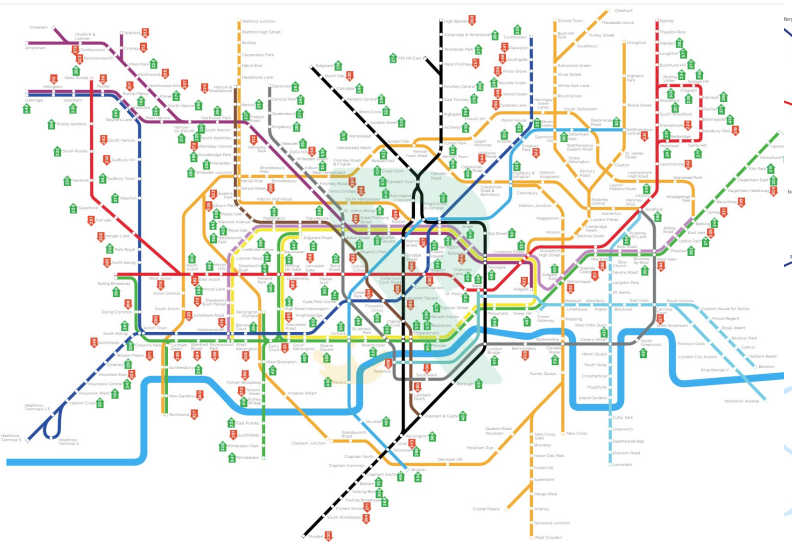
The C4 model does not mandate any specific notation.

- Simon Brown has a suggestion.
- UML can also be used (se the c4model website)

And remember that behavior is equally as important as structure!

So maybe you need some other viewpoints (supported by other diagrams)!






Architectural views

Documenting different viewpoints using 4+1 View Model

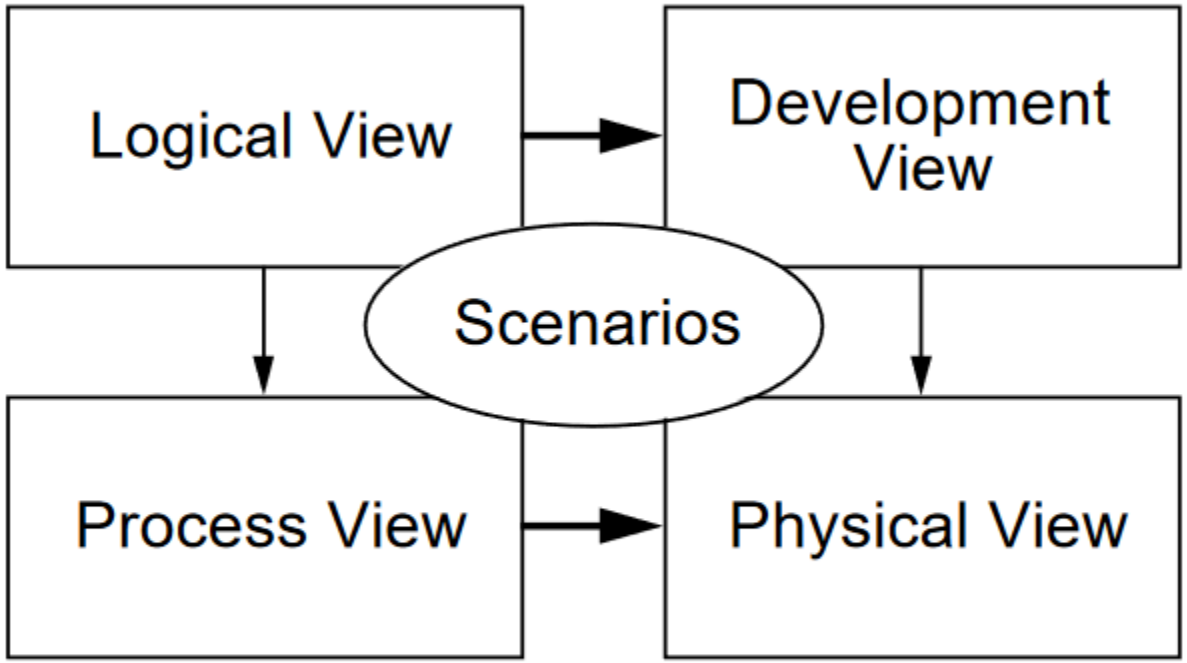
Architectural views

- There is no “one diagram to rule them all”.
- Different “Views” of the system each tells a part of the story.
- Decide what to communicate and choose appropriate diagrams.
  - And decide on the abstraction level.
  - What do you want the reader of the diagram to know?

The 4+1 View Model – Philippe Kruchten

Audience: End-user
Under Study: Functionality

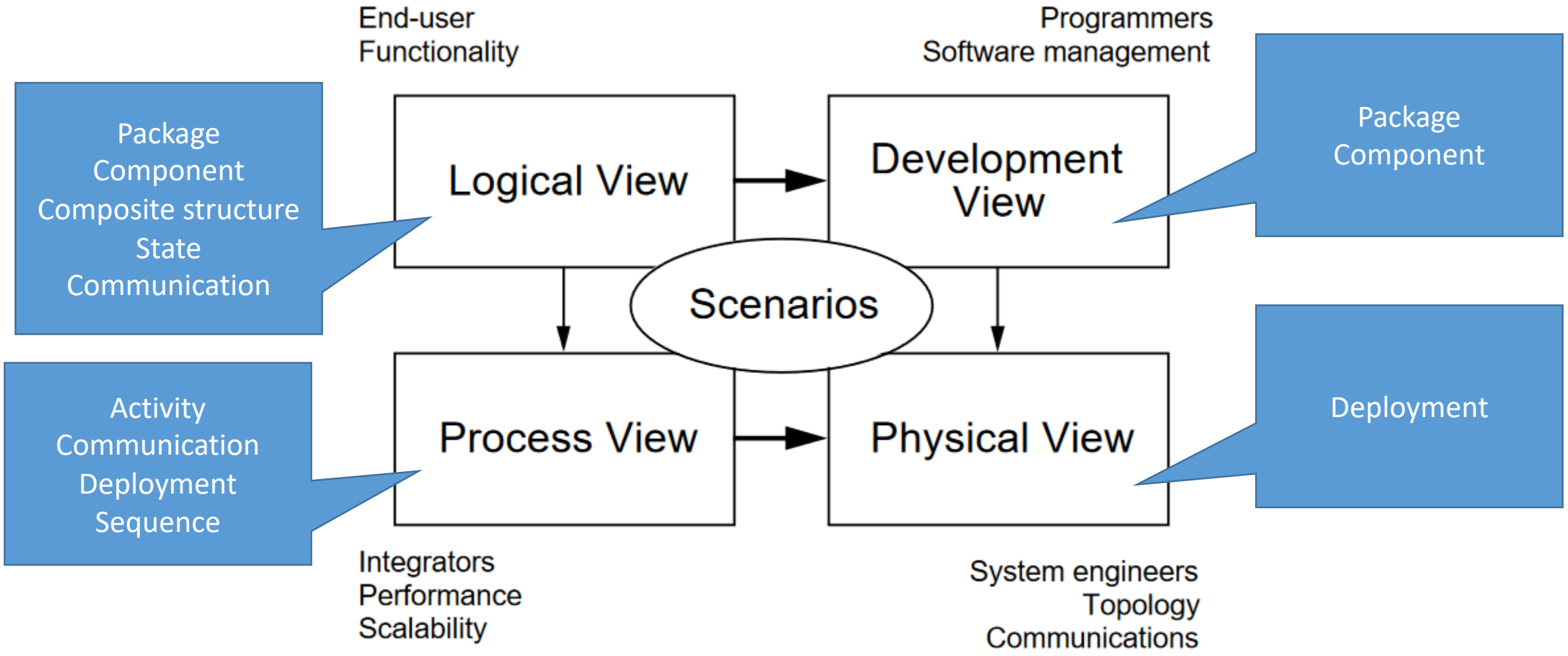
Programmers
Software management



Integrators
Performance
Scalability

System engineers
Topology
Communications

The 4+1 View Model – Philippe Kruchten



Other views

Data

Security

Others...?

Diagrams are not enough

Architectural Knowledge

=

Architectural Design

+

Design Decisions

Eg. C4 and

N+1 views diagrams

Eg. Text

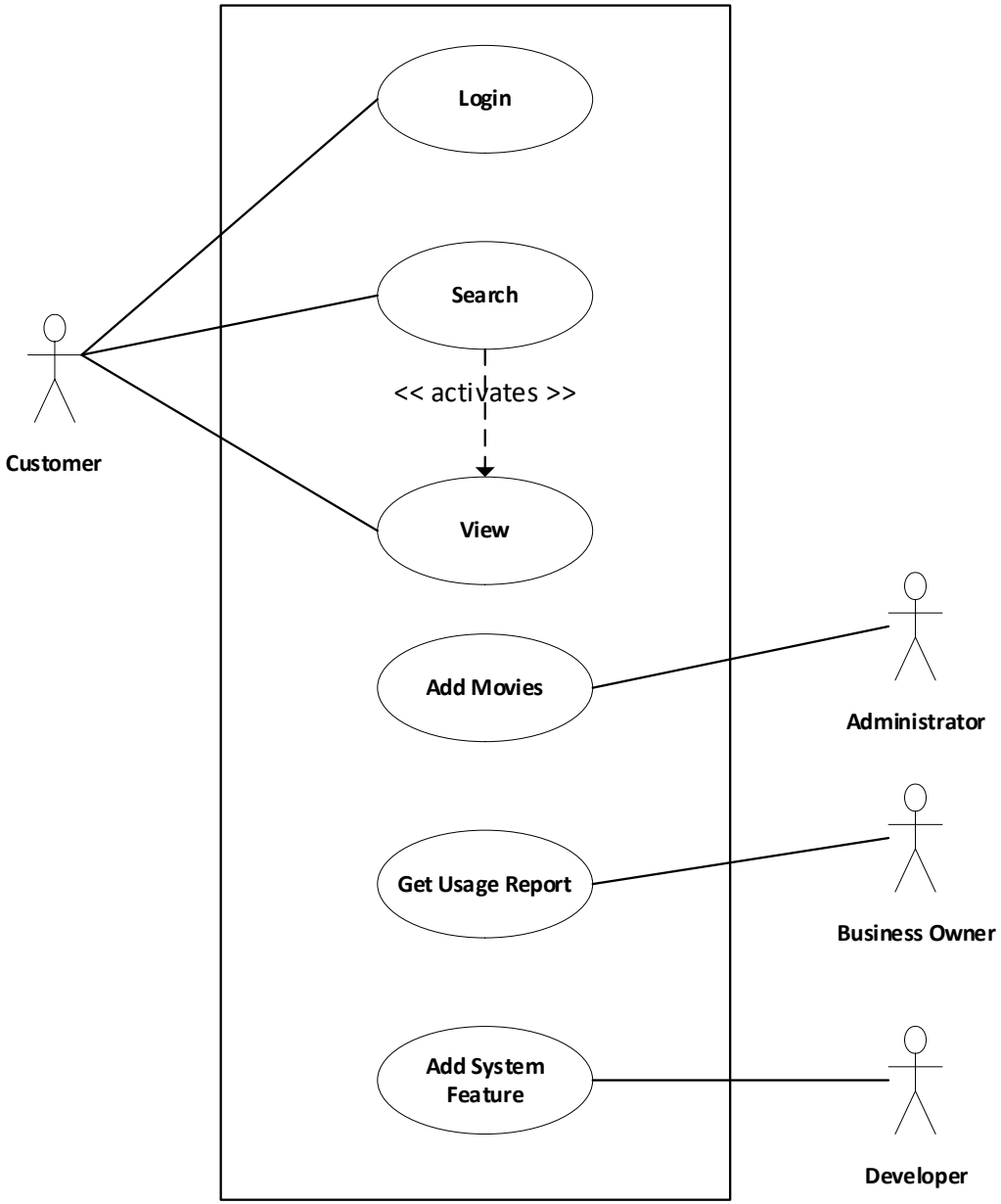
+ Goals and Constraints

Ph. Kruchten, 2009

UML - Structural diagram types

And then: Behavioral diagram types

Use case diagrams



Source: https://www.uml-diagrams.org/multi-layered-web-architecture-uml-package-diagram-example.html

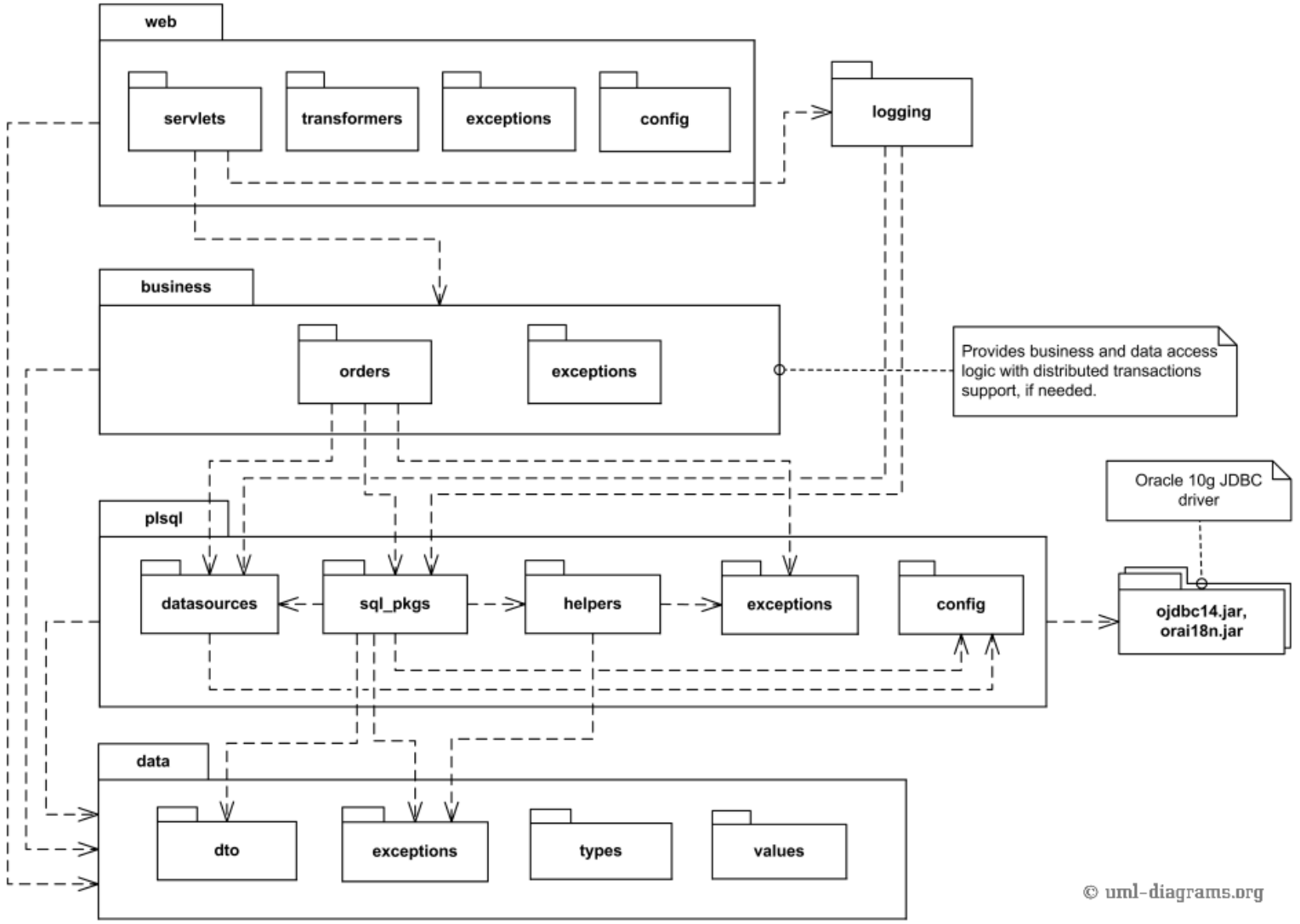
Use case descriptions

USE CASE 5	Buy Goods	
Goal in Context	Buyer issues request directly to our company, expects goods shipped and to be billed.	
Scope & Level	Company, Summary	
Preconditions	We know Buyer, their address, etc.	
Success End Condition	Buyer has goods, we have money for the goods.	
Failed End Condition	We have not sent the goods, Buyer has not spent the money.	
Primary, Secondary Actors	Buyer, any agent (or computer) acting for the customer. Credit card company, bank, shipping service	
Trigger	purchase request comes in.	
DESCRIPTION	Step	Action
	1	Buyer calls in with a purchase request
	2	Company captures buyer's name, address, requested goods, etc.
	3	Company gives buyer information on goods, prices, delivery dates, etc.
	4	Buyer signs for order.
	5	Company creates order, ships order to buyer.
	6	Company ships invoice to buyer.
	7	Buyers pays invoice.

https://pja.mykhi.org/0sem/INN/sorcersoft.org/io/uml/UML-borland-UCDs.html

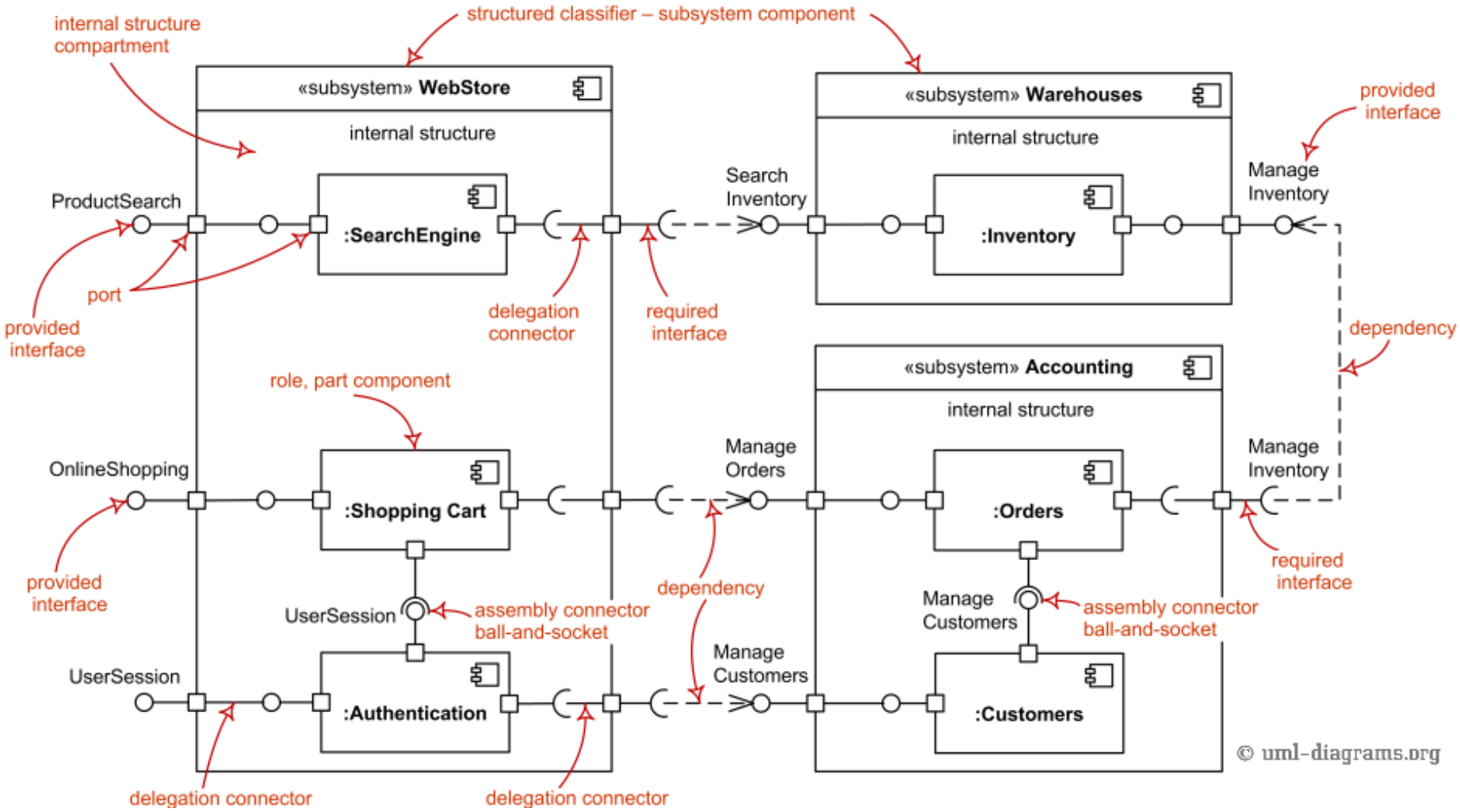
EXTENSIONS	Step	Branching Action
	3a	Company is out of one of the ordered items: 3a1. Renegotiate order.
	4a	Buyer pays directly with credit card: 4a1. Take payment by credit card (use case 44)
	7a	Buyer returns goods: 7a. Handle returned goods (use case 105)
SUB-VARIATIONS		Branching Action
	1	Buyer may use phone in, fax in, use web order form, electronic interchange
	7	Buyer may pay by cash or money order check credit card

Package diagrams



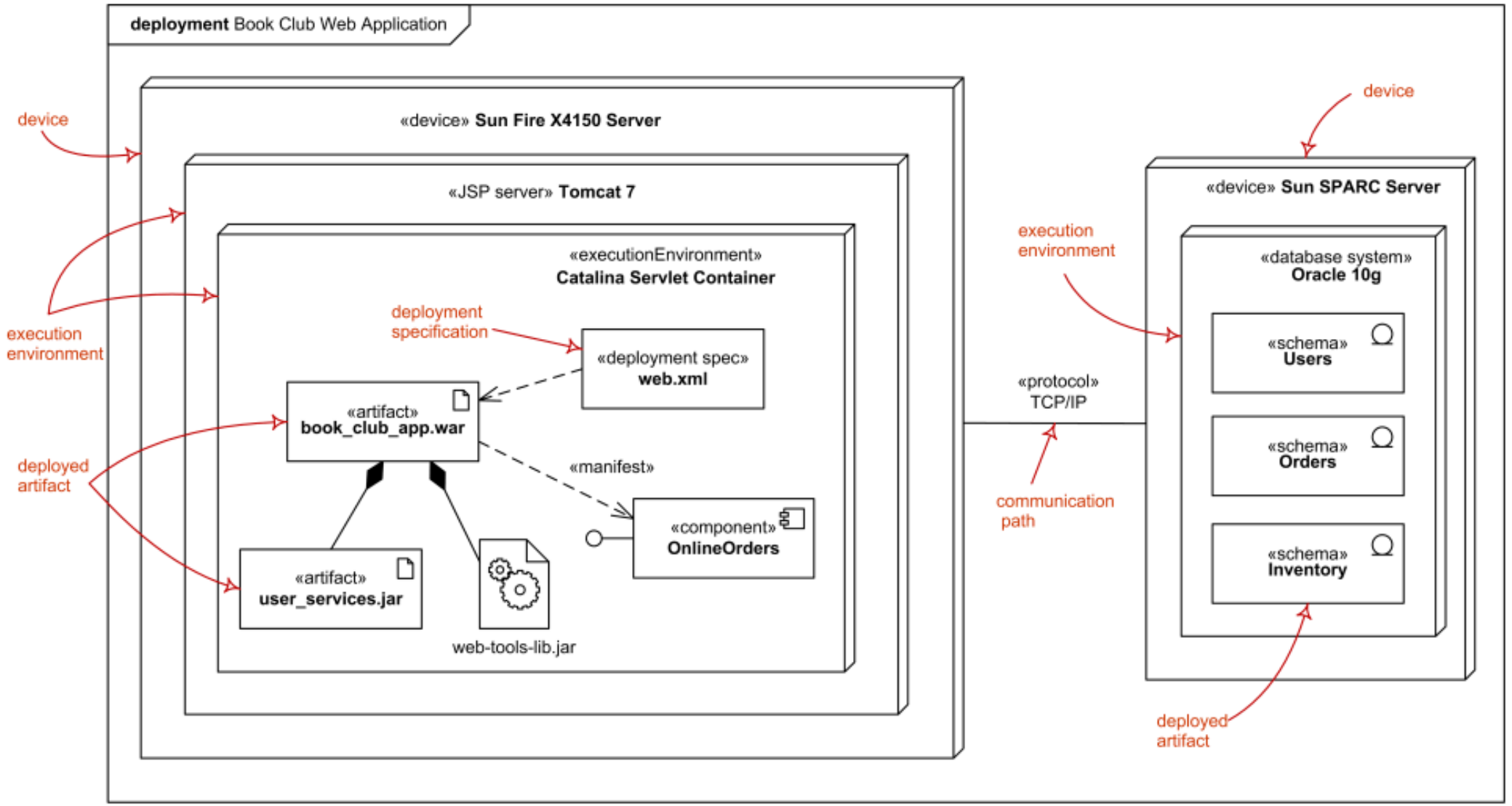
Source: https://www.uml-diagrams.org/multi-layered-web-architecture-uml-package-diagram-example.html

Component diagrams



Source: https://www.uml-diagrams.org/component-diagrams.html

Deployment diagrams



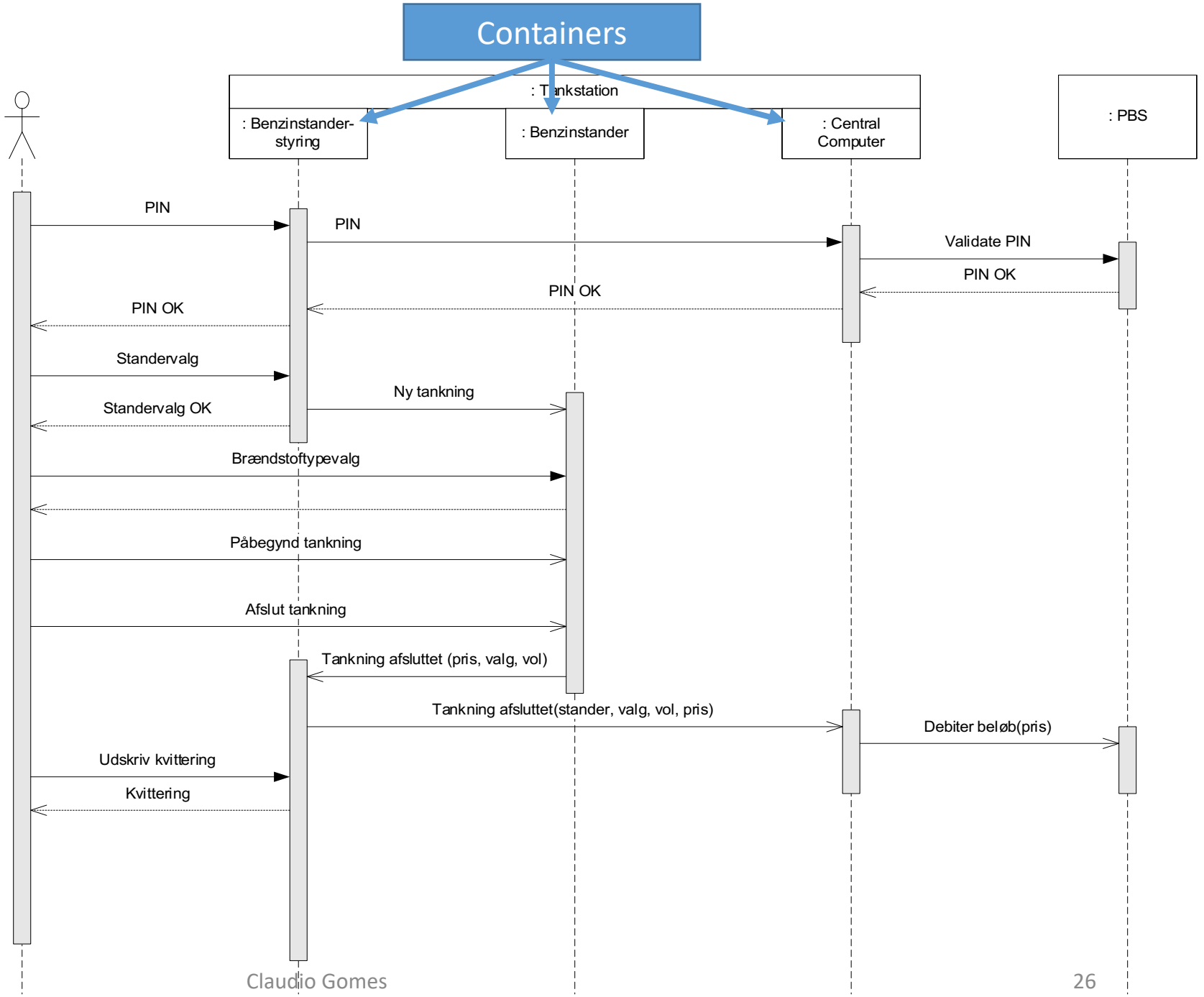
Source: https://www.uml-diagrams.org/deployment-diagrams-overview.html

UML - Behavioral diagram types

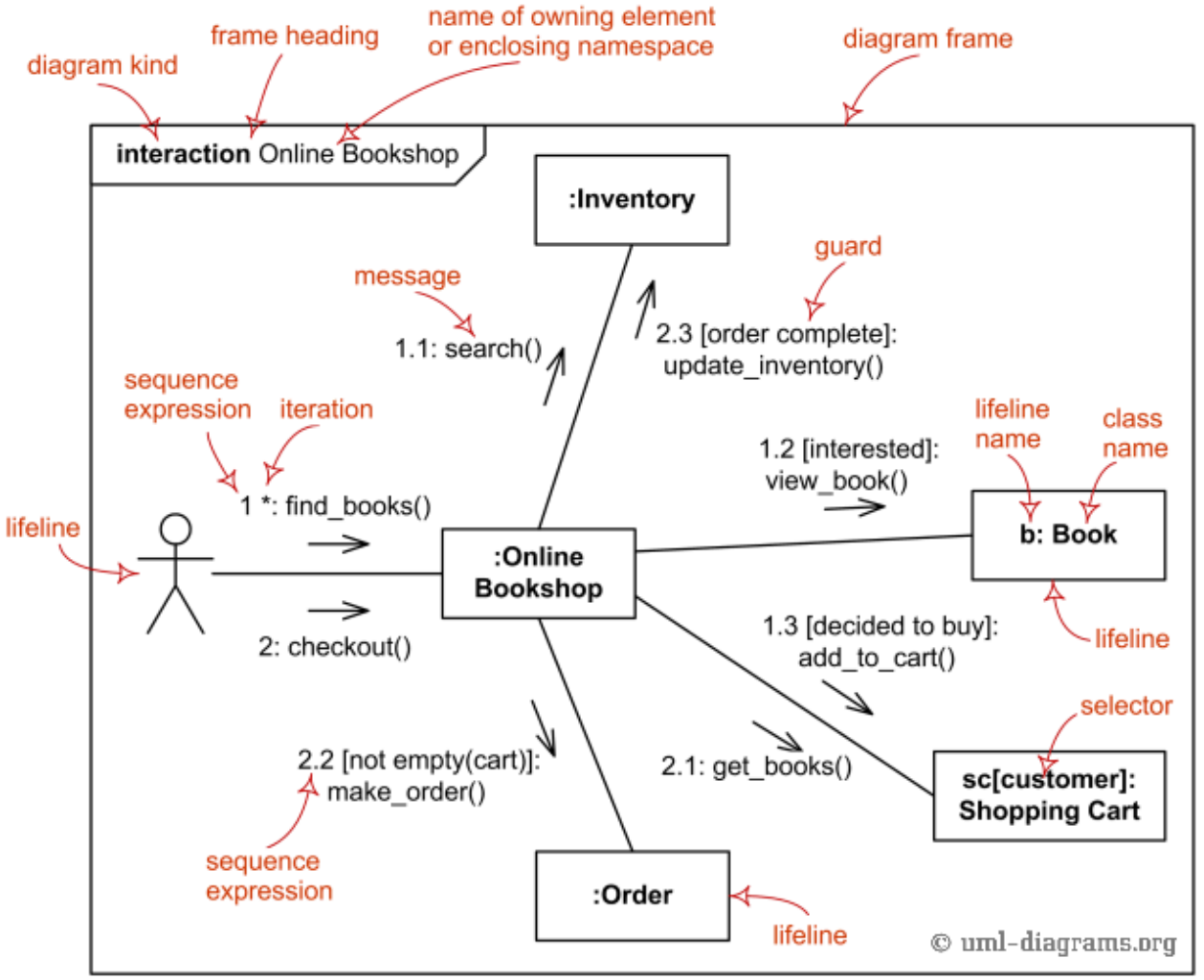
Sequence diagrams

- Sequence diagrams can be used at different levels of abstraction
  - Between system and context
  - Between containers
  - Between components
  - Between classes

Sequence diagrams



Communication diagrams



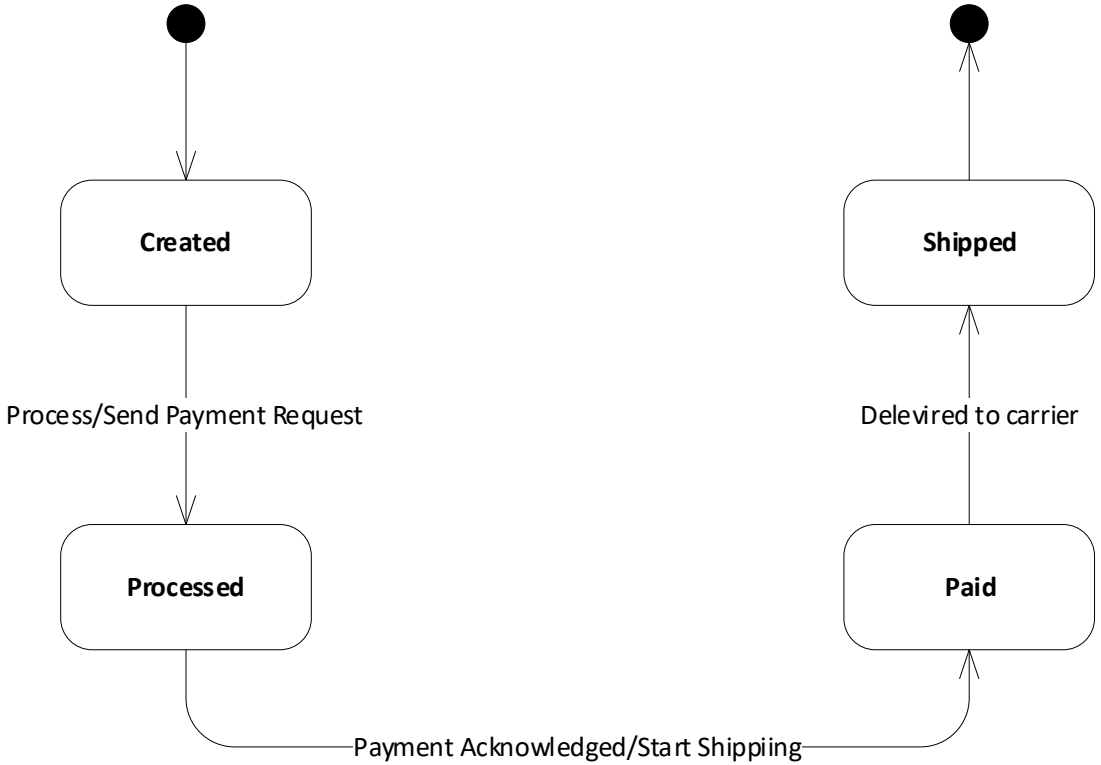
Source: https://www.uml-diagrams.org/communication-diagrams.html

State Machine diagrams

- Can be used at different abstraction levels
  - To describe the conceptual state of a Domain class that the user and other non-technical stakeholders can understand
  - As a Design pattern for a process or a class at the implementation level
    - Especially good for asynchronous/event driven activities, as between containers

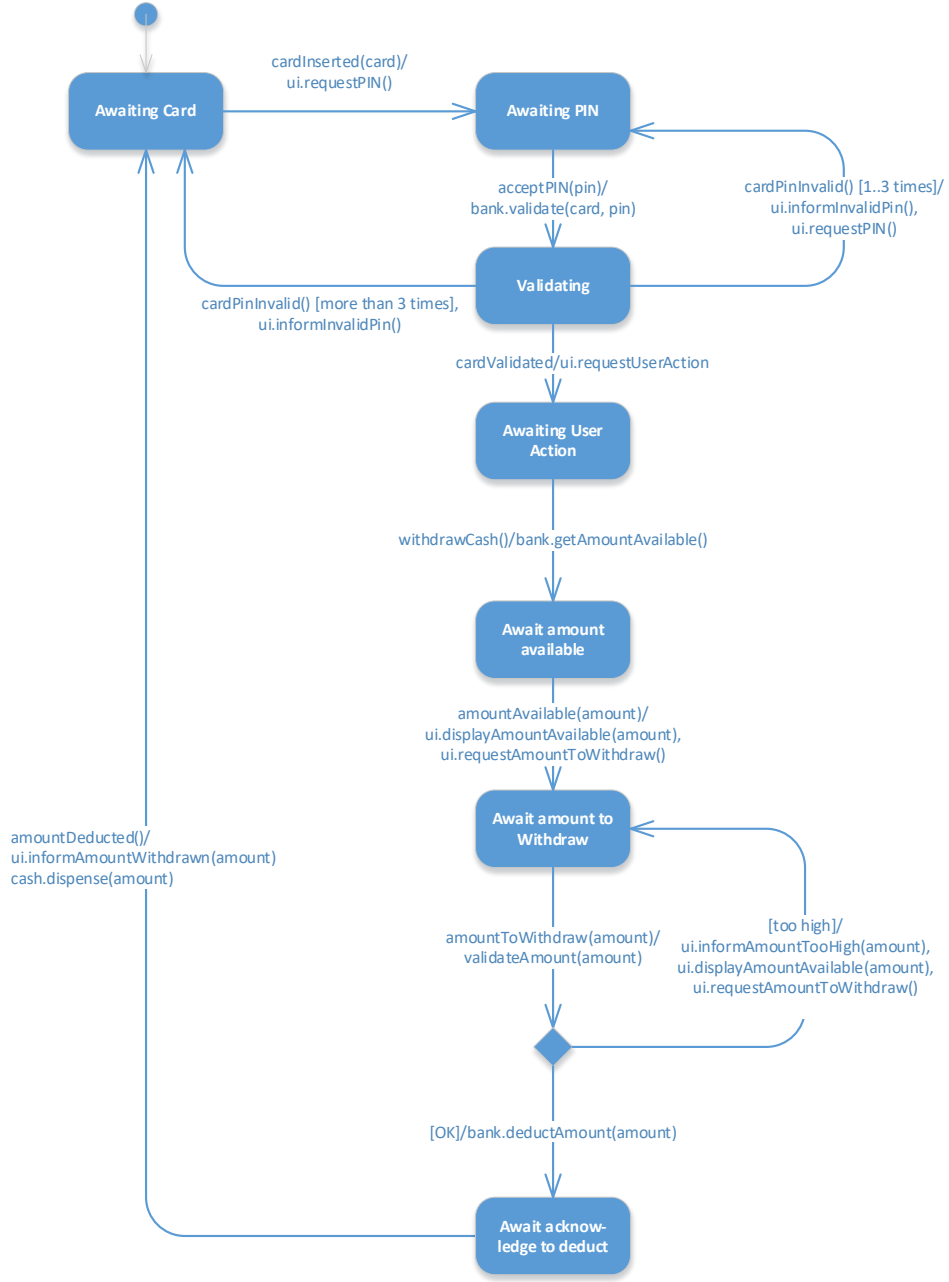
State Machine diagrams

- Conceptual states for a domain class: Order

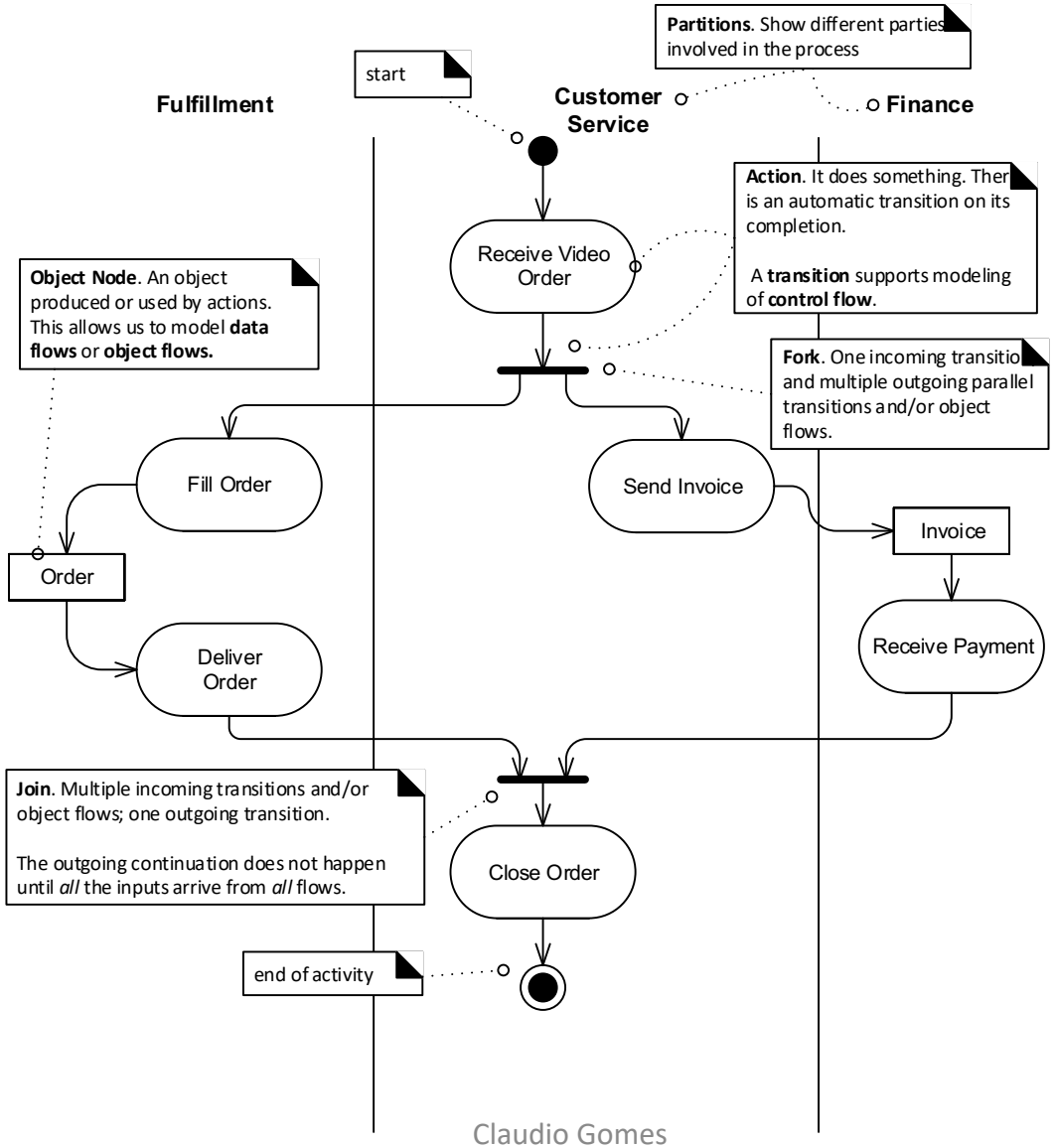


State Machine diagrams

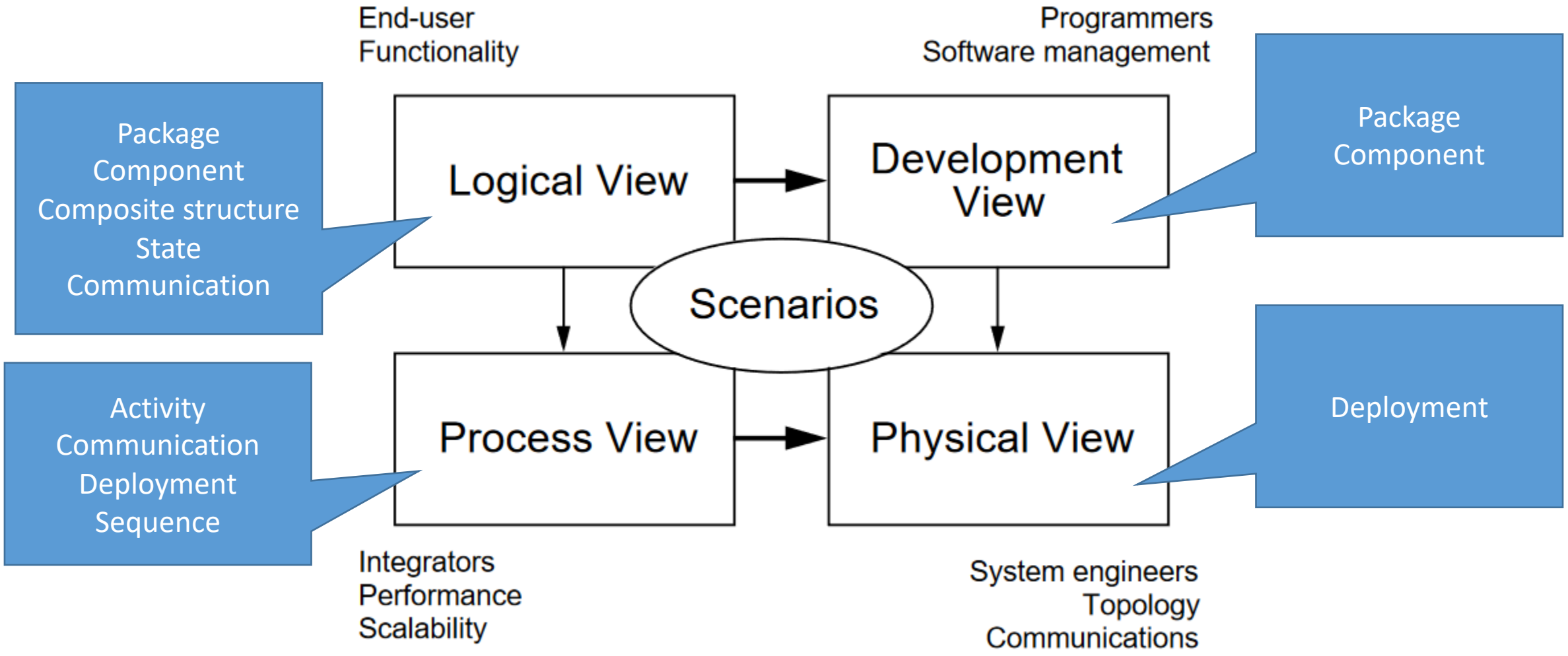
- Implementation STM for a transaction communicating with other containers



Activity diagrams



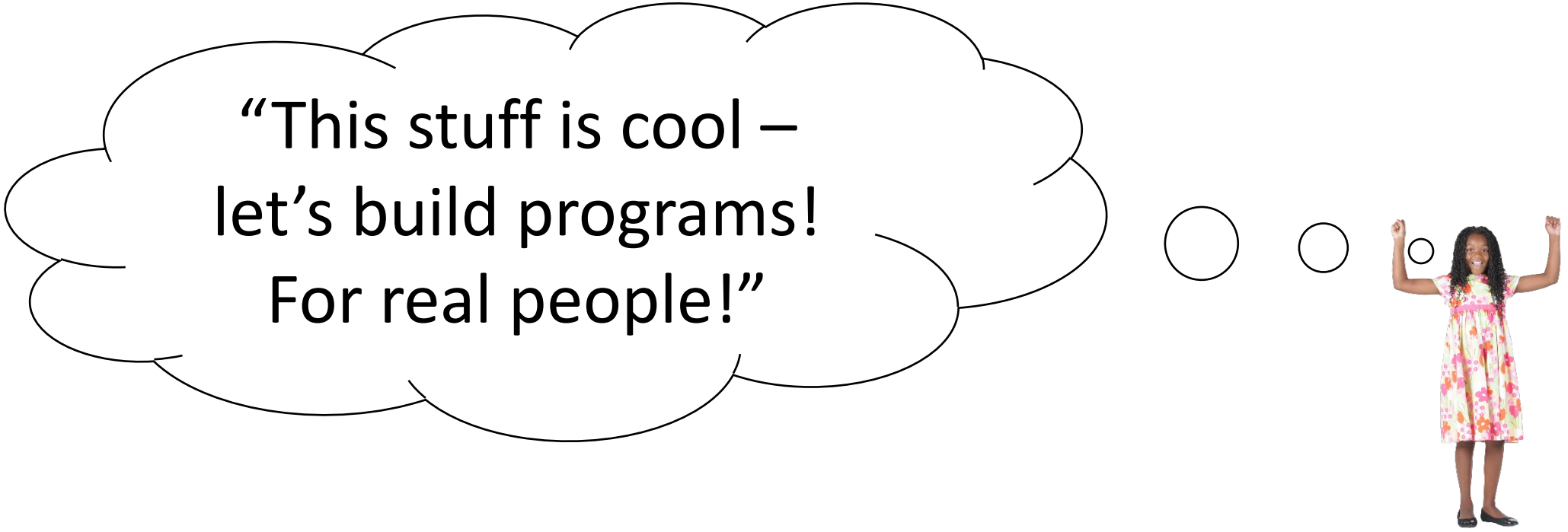
The 4+1 View Model – Philippe Kruchten



How do you become good at architecture?

Phases in the Developer's Life

1. The enthusiastic developer



[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

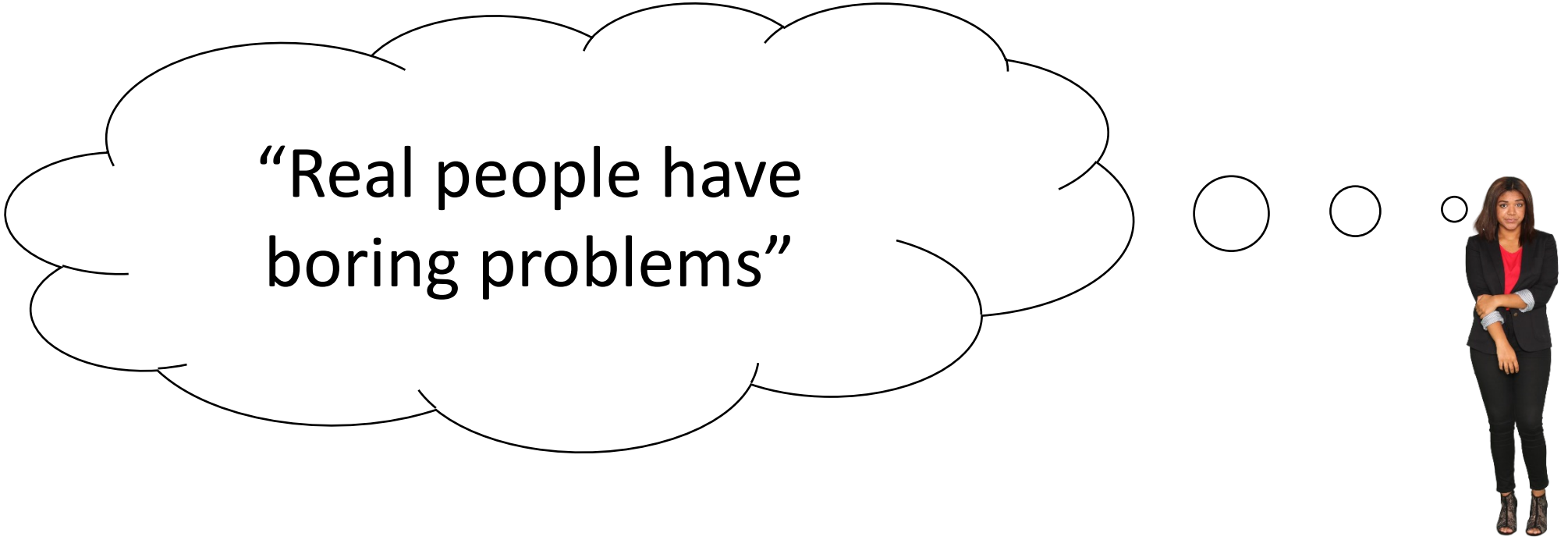
1. The enthusiastic developer

Create Customer	Create Product	Create Order
Find Customer	Find Product	Find Order
List Customers	List Products	List Orders
Edit Customer	Edit Product	Edit Order
Delete Customer	Delete Product	Delete Order

[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

I. The enthusiastic developer



II. The disillusioned developer

[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

II. The disillusioned developer

Create Customer	Create Product	Create Order
Find Customer	Find Product	Find Order
List Customers	List Products	List Orders
Edit Customer	Edit Product	Edit Order
Delete Customer	Delete Product	Delete Order

[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

II. The disillusioned developer

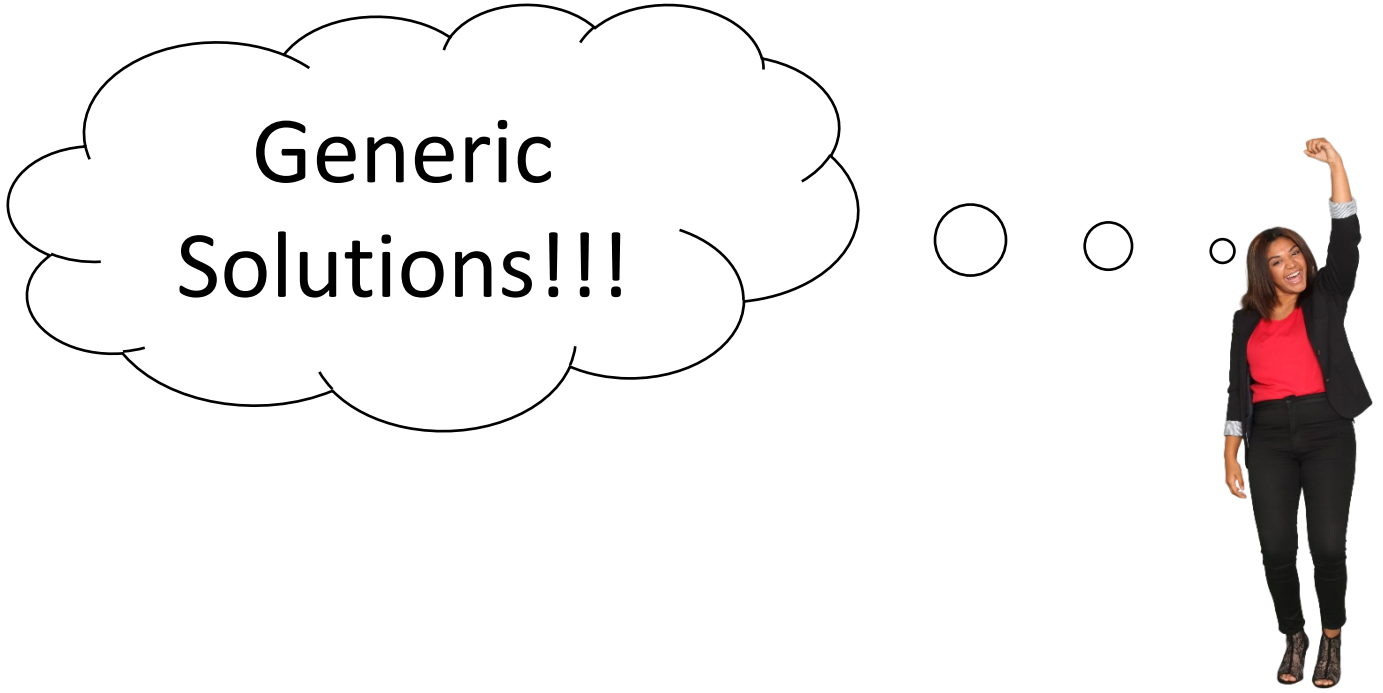
Create Thing

Find Thing

List Thing

Edit Thing

Delete Thing



III. The enthusiastic architect

[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

III. The enthusiastic architect

KISS

YAGNI

Lean

Minimable viable product

Story focus

IV. The disillusioned architect

[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

Phases in the Developer's Life

V. The "wise" architect

Question: *

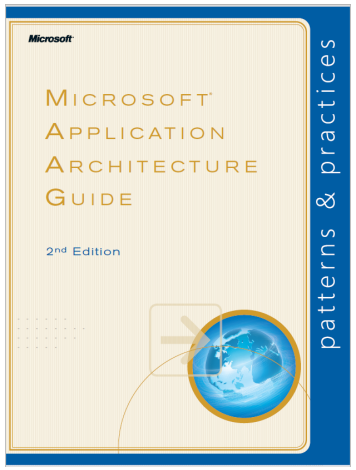
Answer: It depends.

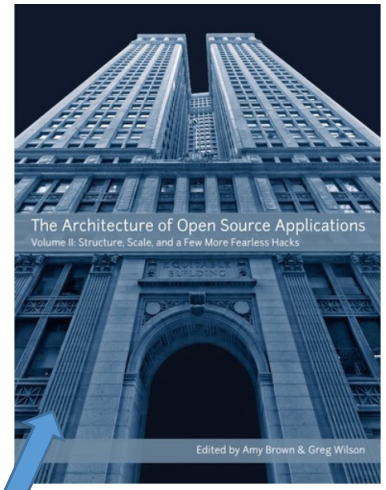
[1] Stefan Tilkov: Why software architects fail – and what to do about it - Craft Conference 2019. https://www.youtube.com/watch?v=AkYDsiRVqno&ab_channel=CraftHubEvents

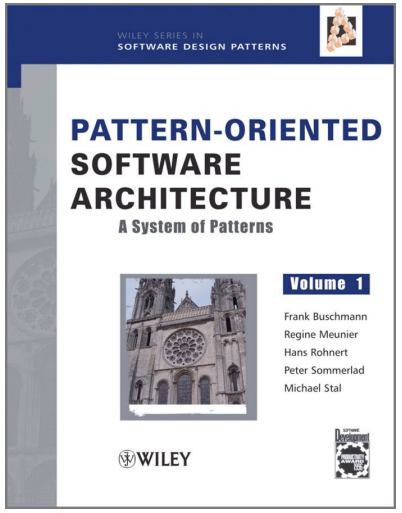
Practice

WAX ON,
WAX OFF.

Learn from others







http://aosabook.org/en/index.html

Read and watch

Martin Fowler

https://martinfowler.com/

Simon Brown

http://www.codingthearchitecture.com/

Scott Ambler

http://www.agilemodeling.com/

SA Research

- Domain Specific Languages and Model Driven Engineering
- Consistency Management
- Engineering of Digital Twins

Your turn again!

Draw the architecture

Pair with someone, who is not working
on the same project as you.

You have 10 minutes to
explain the software architecture
of your project to the other person.
Then switch roles.

Use paper and pencil (or a whiteboard).
PC's and pre-made diagrams are forbidden!

AARHUS UNIVERSITY