

Ardeim

*(Ar)chitecture, (De)sign &
(Im)plementation*

Copyright:
SivaramaSundar, Karthik Kalkur

Thanks to:
Rashmi Subbanna

Credits:
revealjs, striptthis (kesiev)

Heads UP!

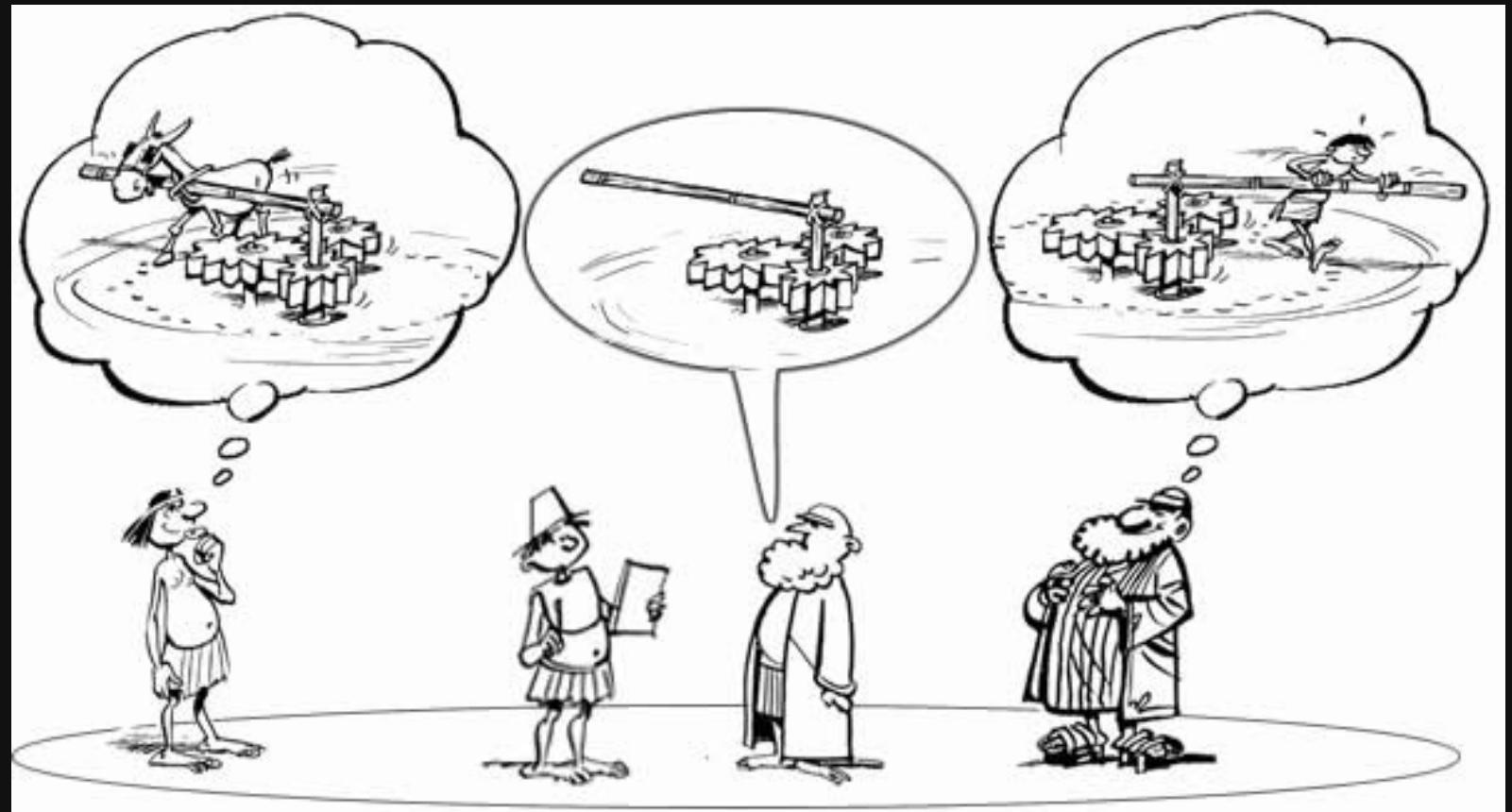


details

Raise relevant questions to understand the business benefits!

Heads UP!

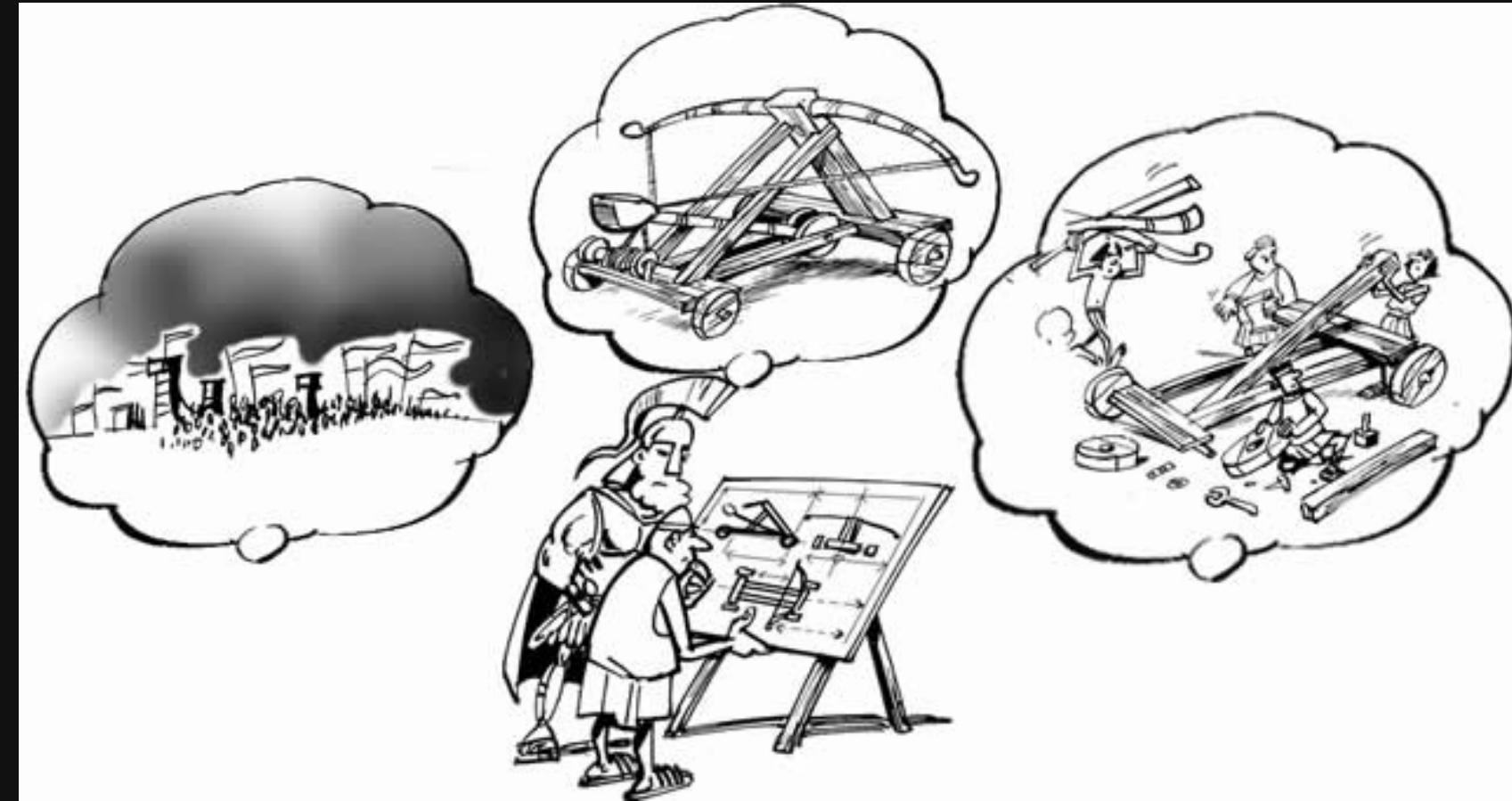
What you think is what you get!



Specific & relevant details helps you think through a better solution!

Heads UP!

Prespectives & Concerns Differ!



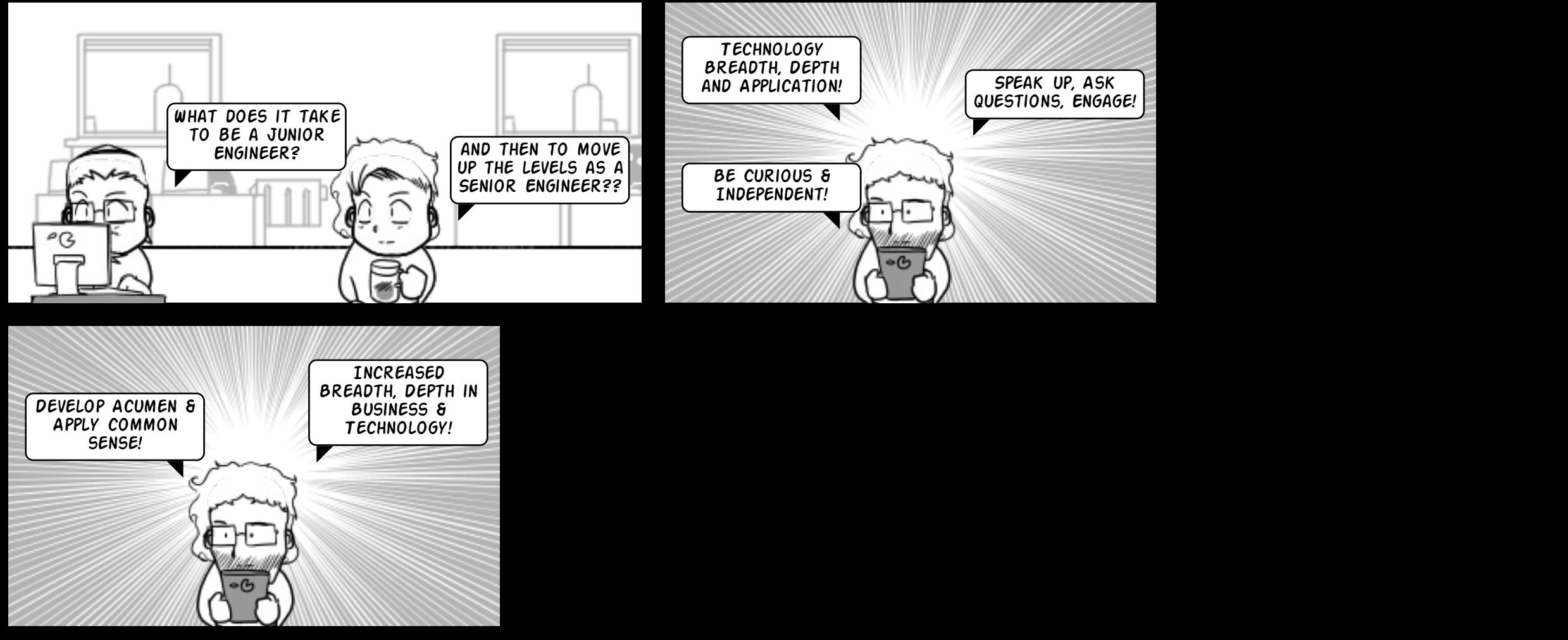
Understand Business Needs vs Functionalities vs Implementation
from the Stakeholder Prespective!

Context

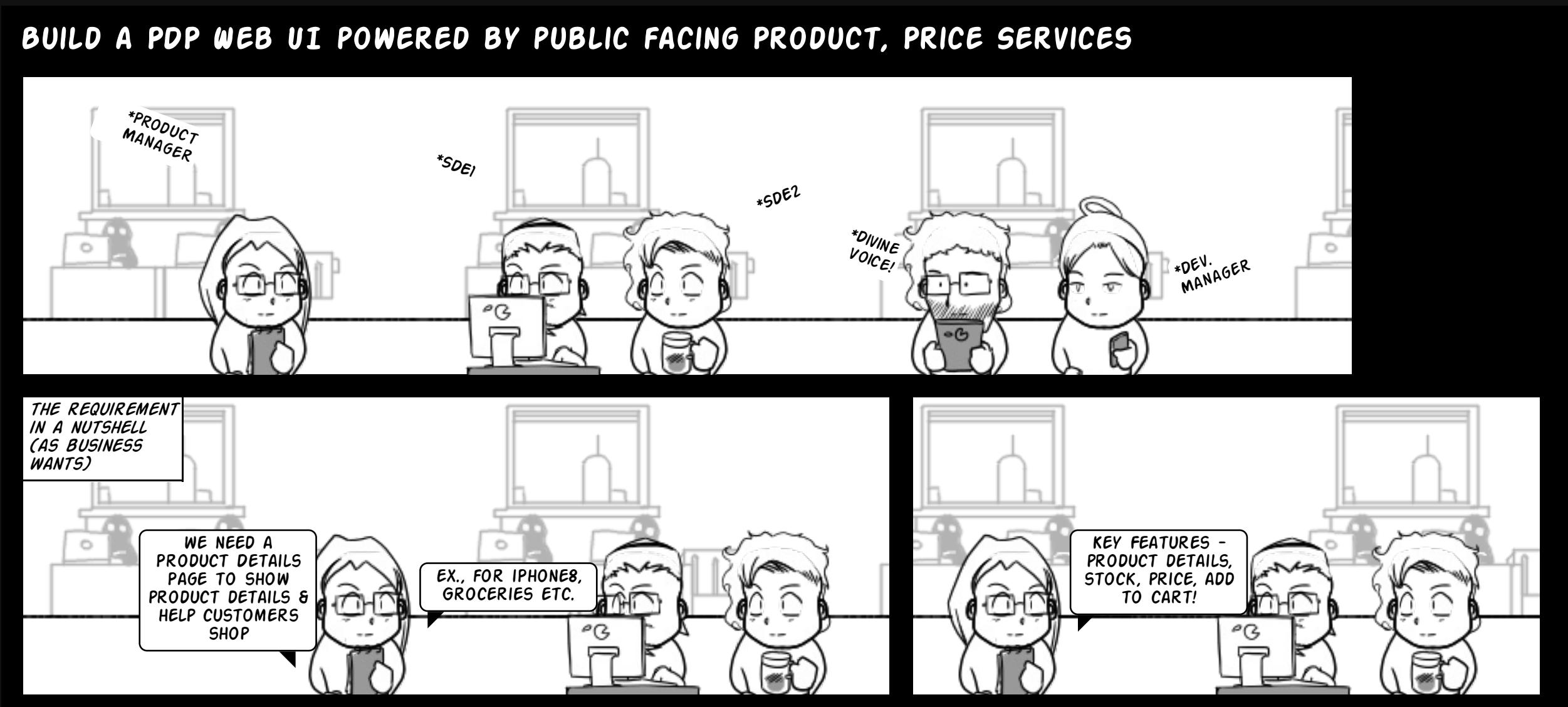


Context

THE ENGINEER'S DILEMMA



Context



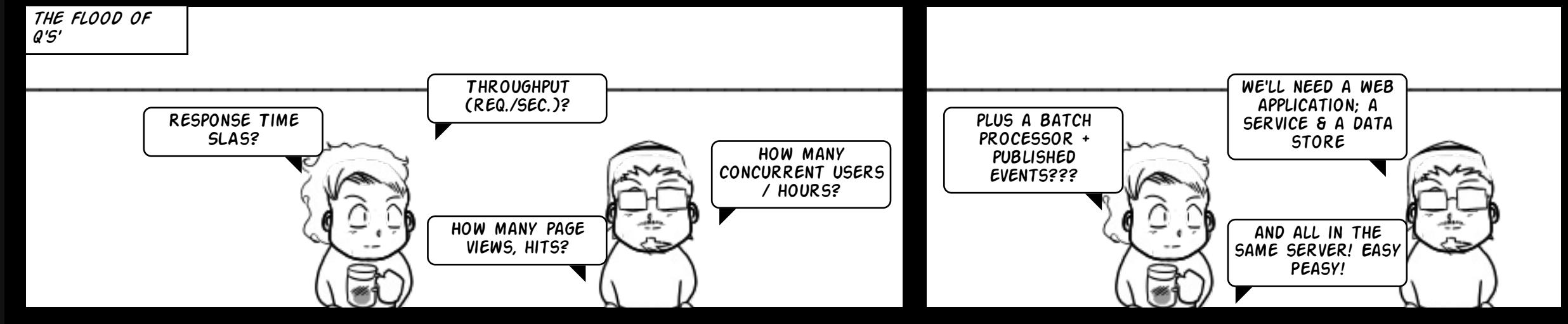
Context

... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES



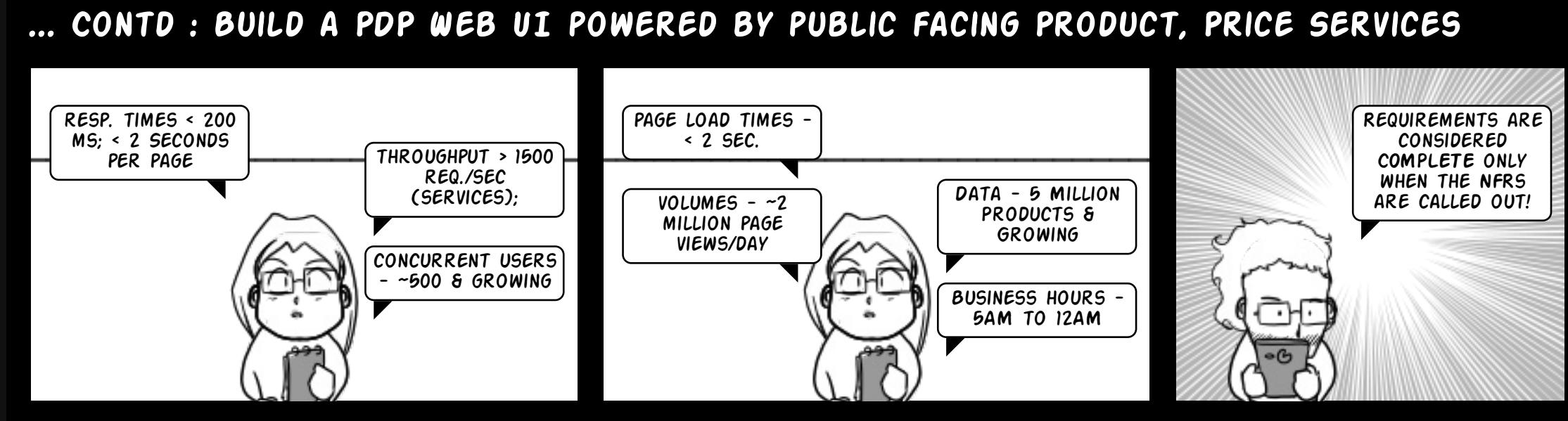
Context

... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES



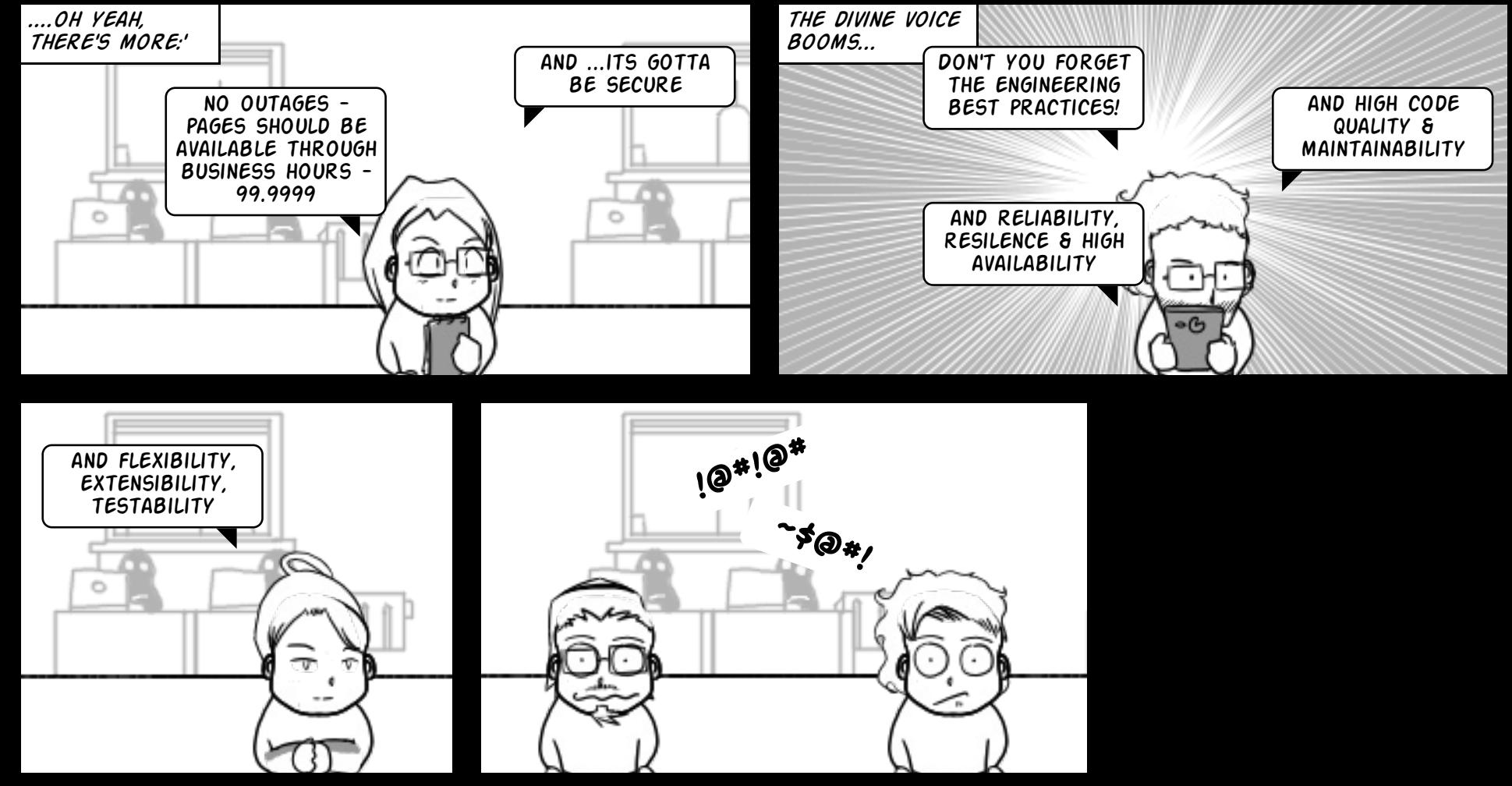
Context

... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES

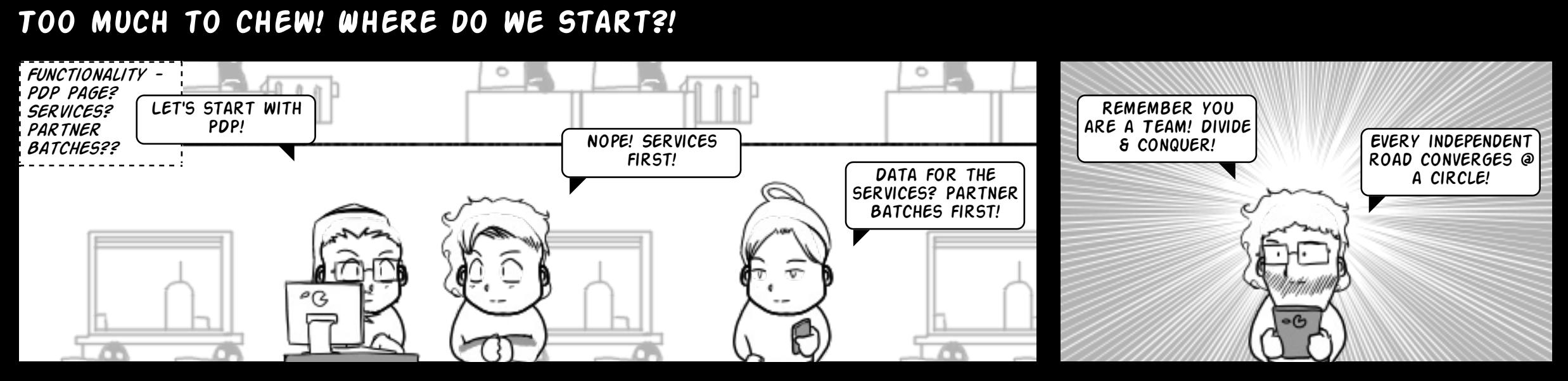


Context

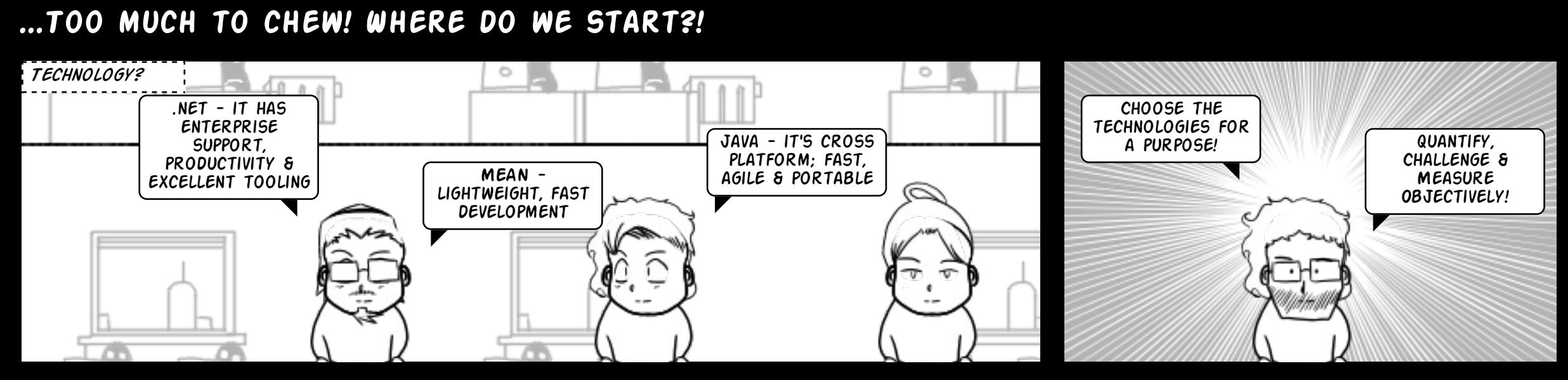
WAIT A MIN ... THERE'S MORE



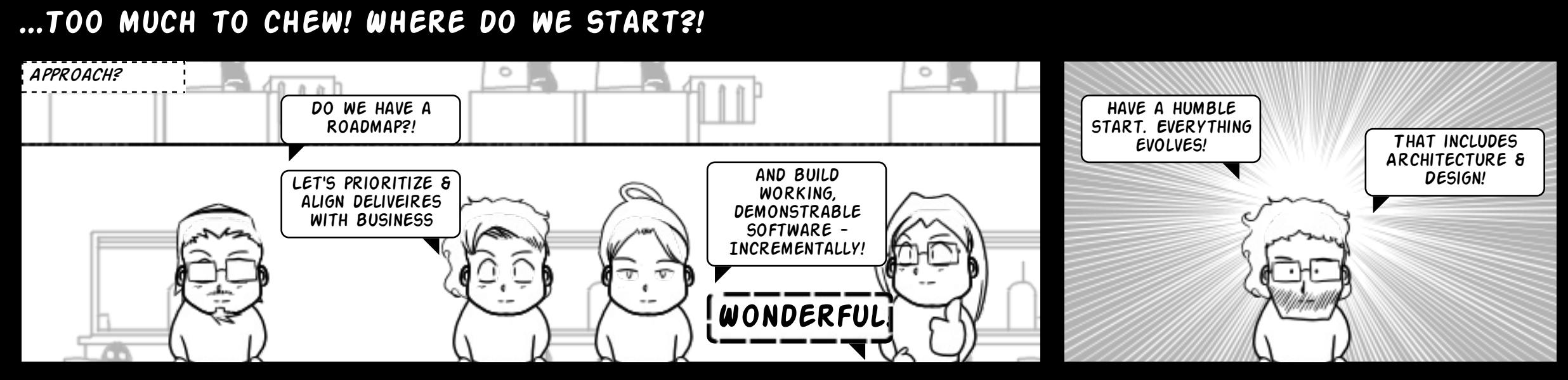
What next!?



What next!?



What next!?

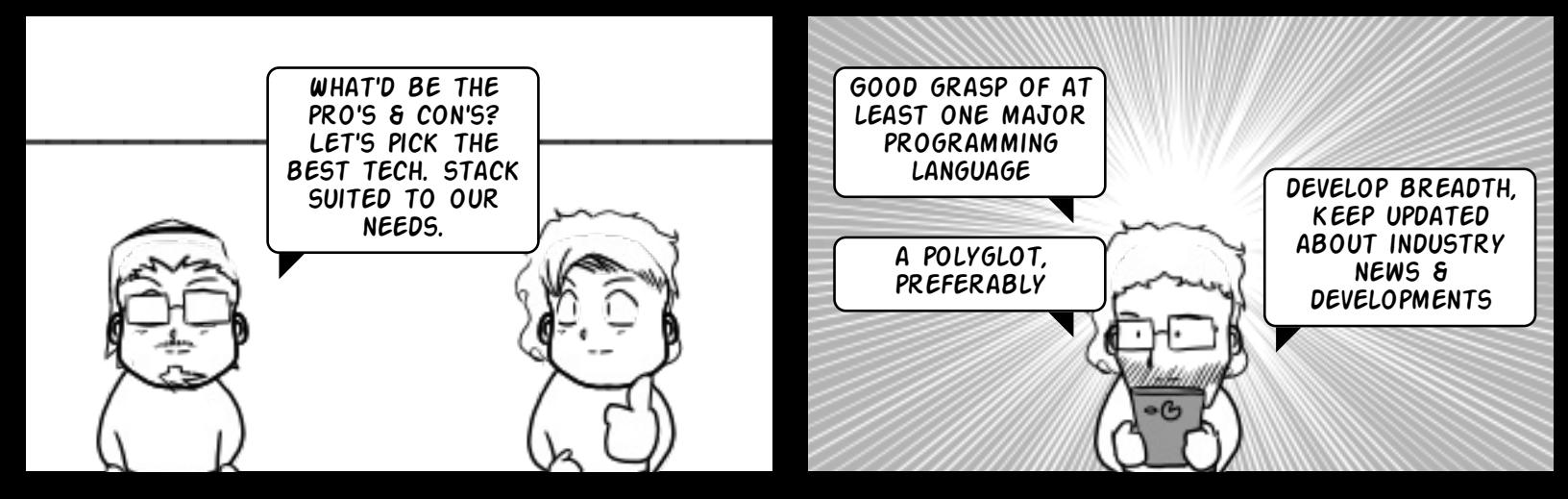


What next!?

Links: Domain Driven Design - Eric Evans

What next!?

...TOO MUCH TO CHEW! WHERE DO WE START?!



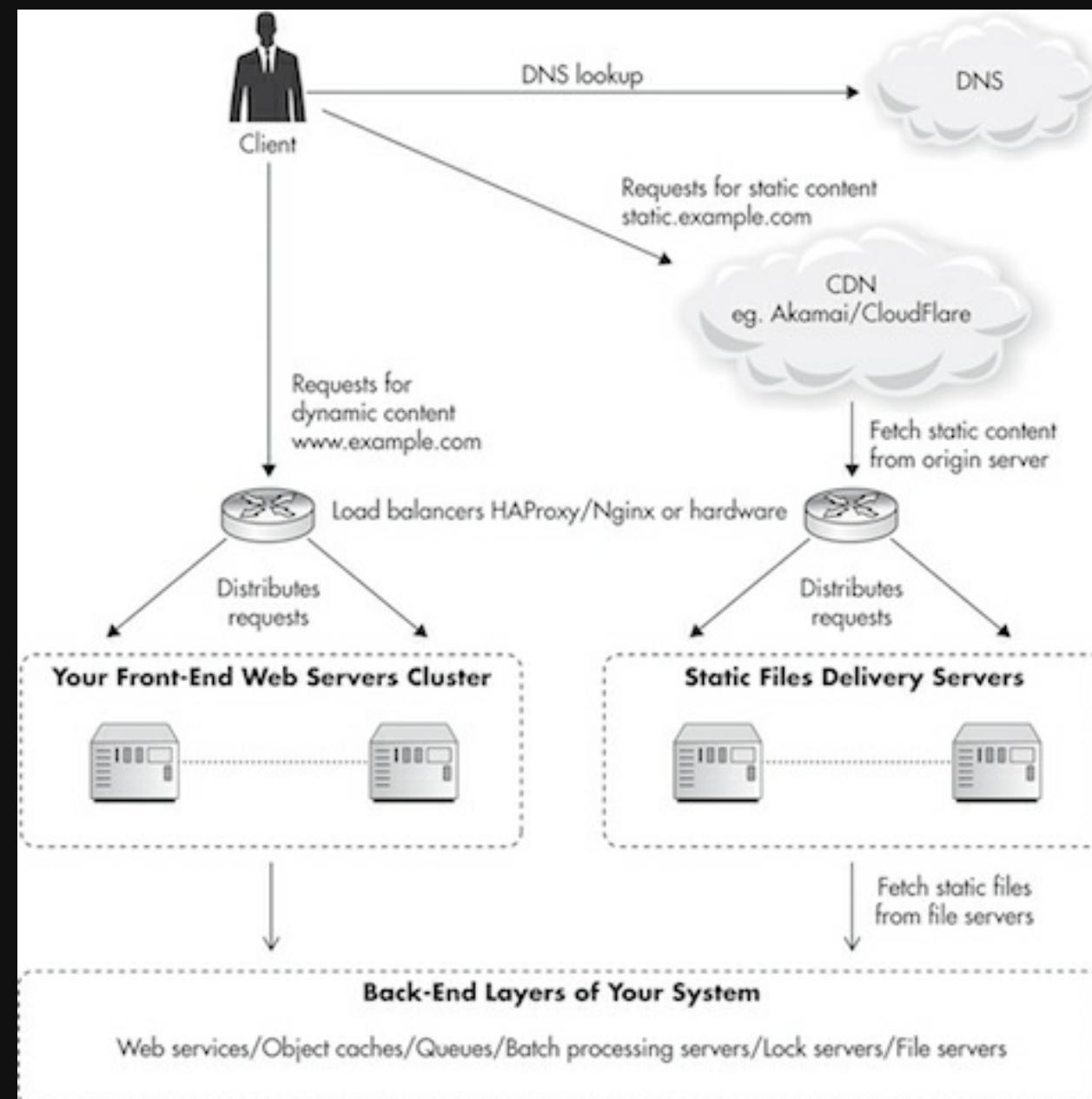
What next!?

...CONTD!



What next!?

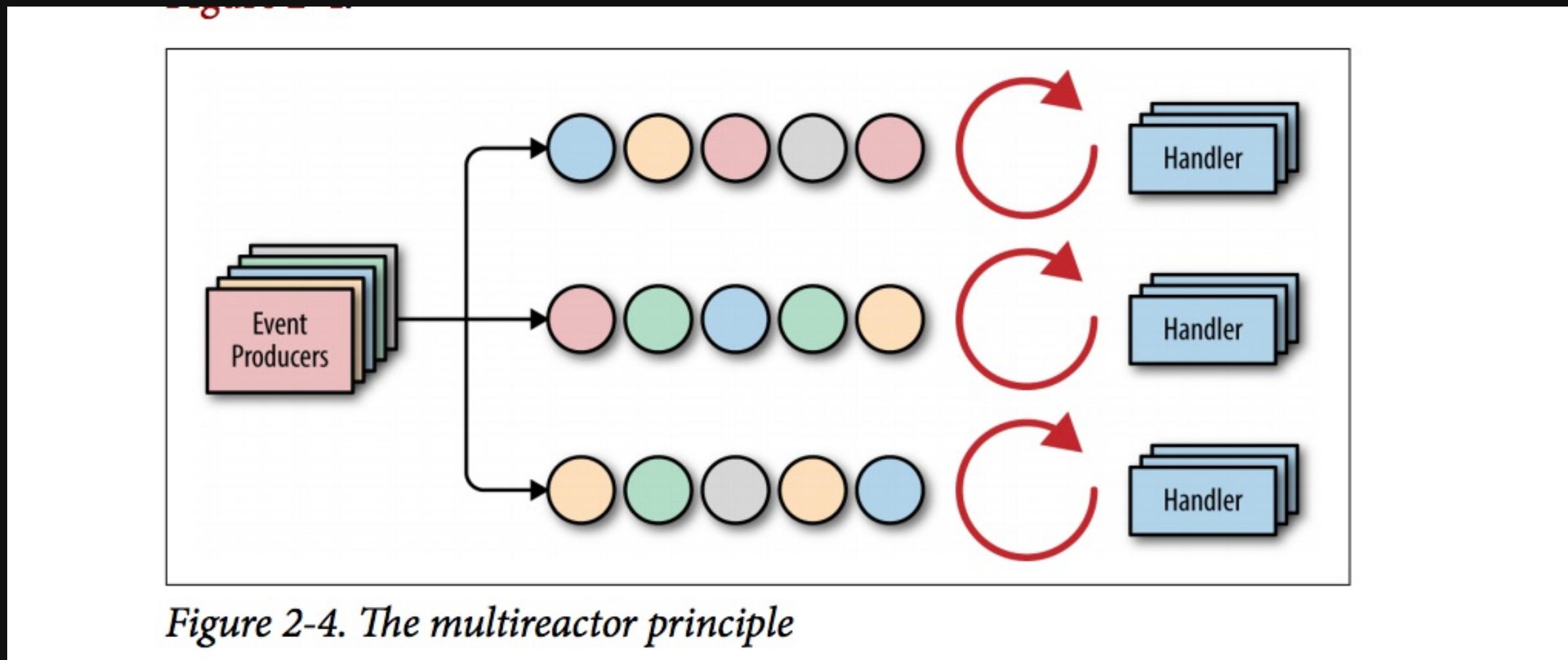
Which Technology to choose for App Layer?



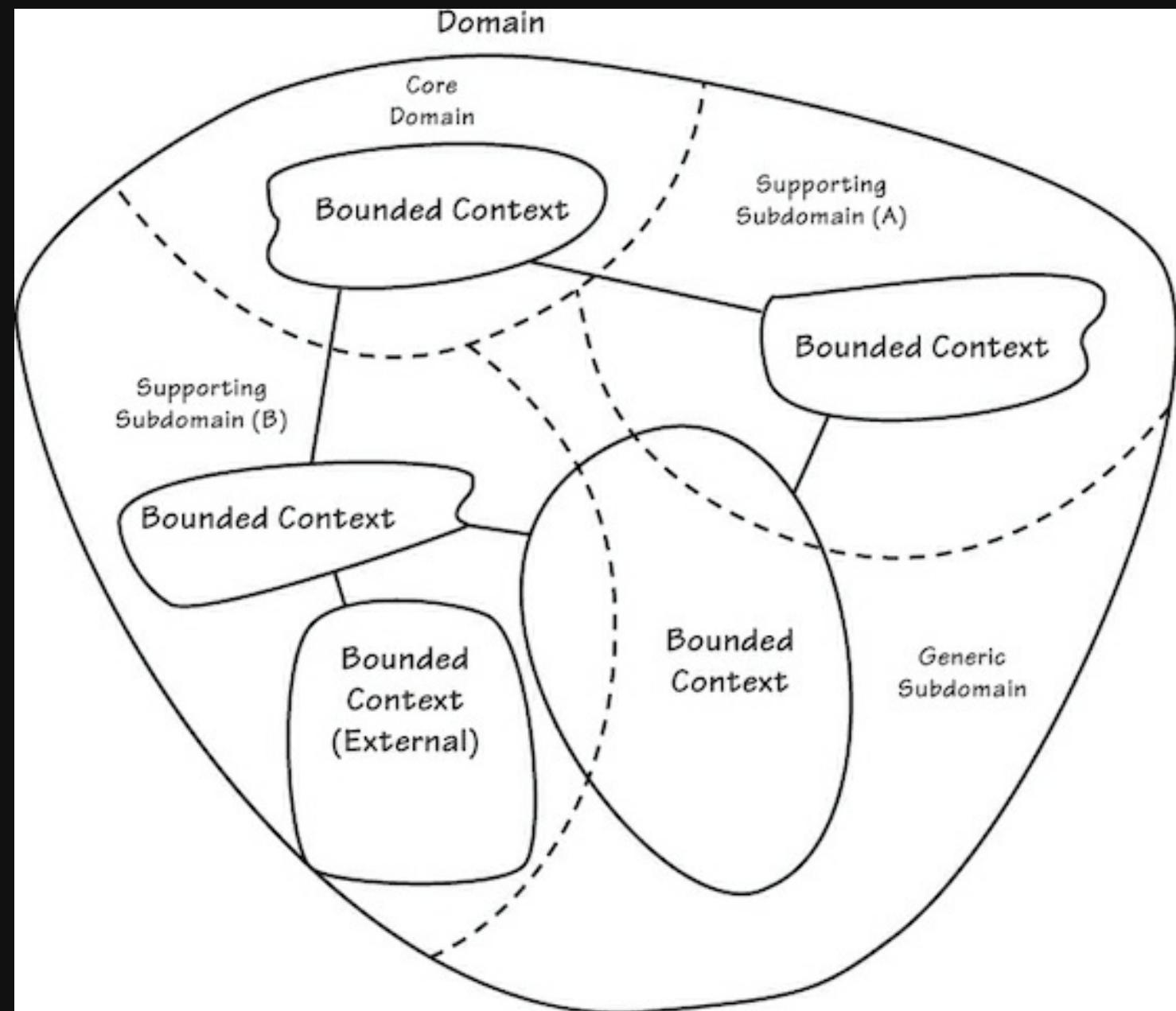
NGINX based on event loop scales well also provides load balancing and reverse proxy

What next!?

Choose the Framework with right Threading Model
Multiteactor Pattern Node.js or Vert.x?

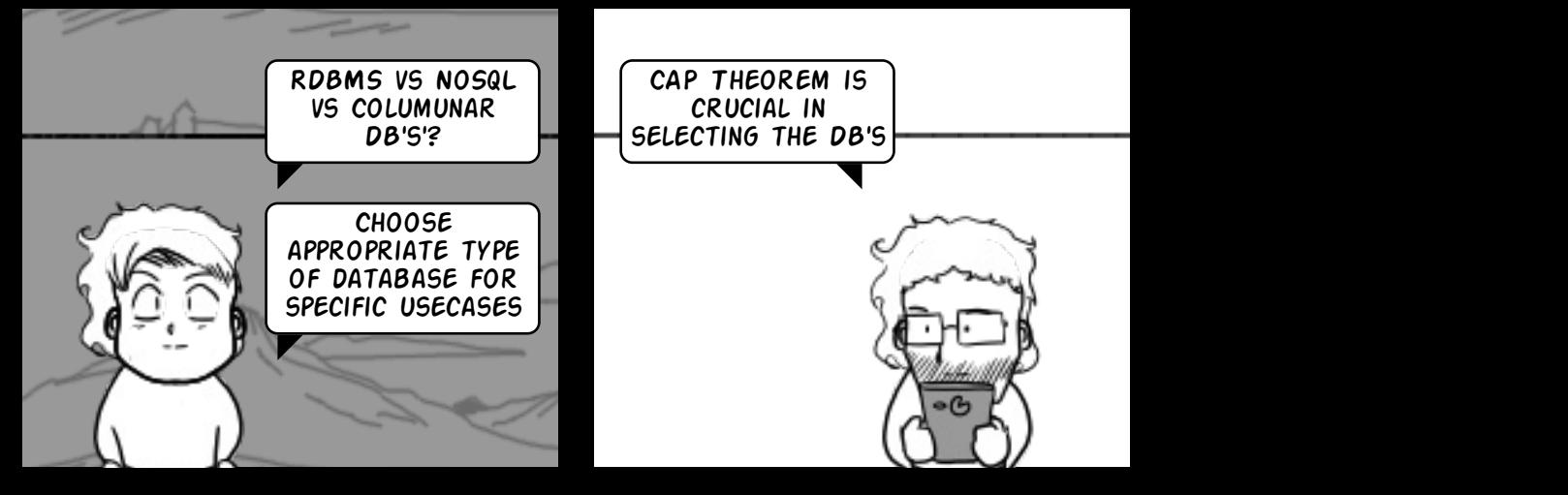


What next!?



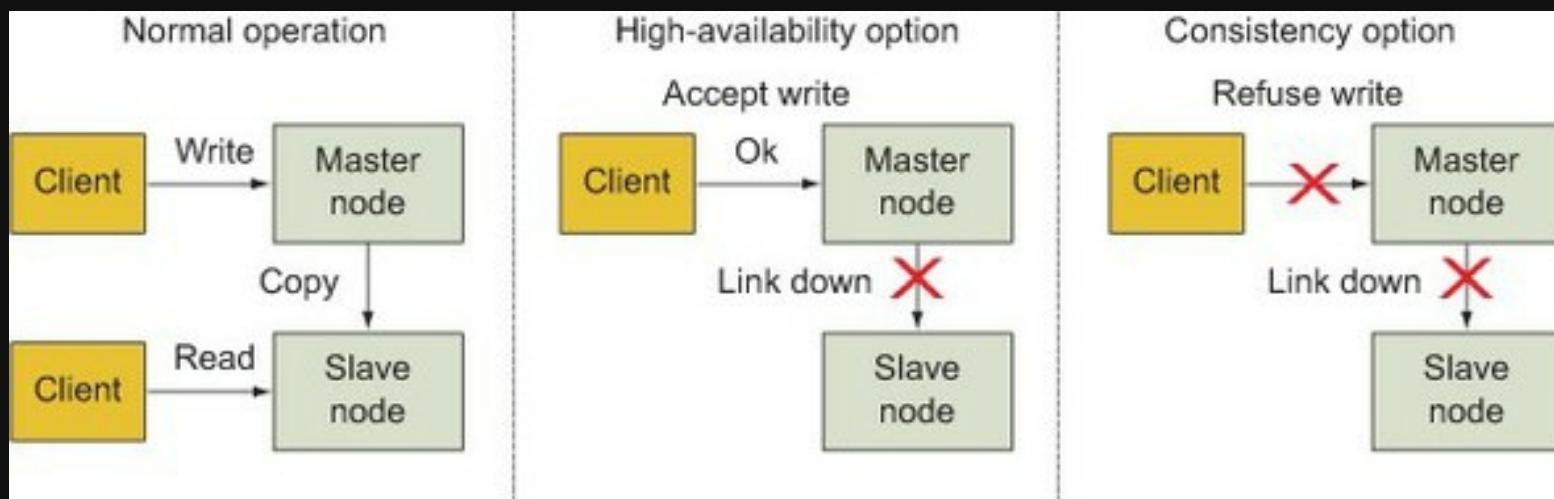
What next!?

CHOOSE THE TECHNOLOGIES, FRAMEWORKS & TOOLS FOR DB LAYER



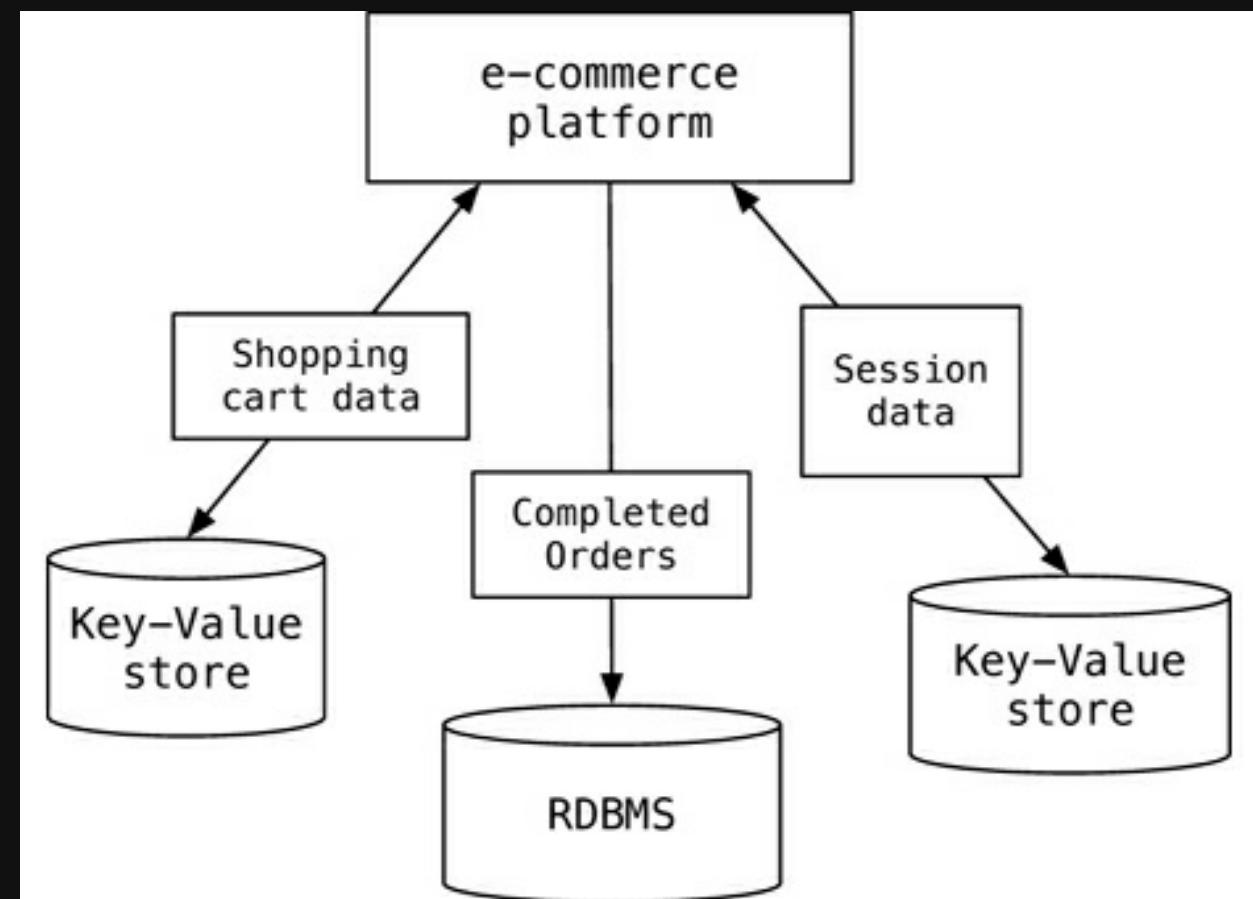
What next!?

CAP Therom Availability , Constitancy, Partition Tolerance



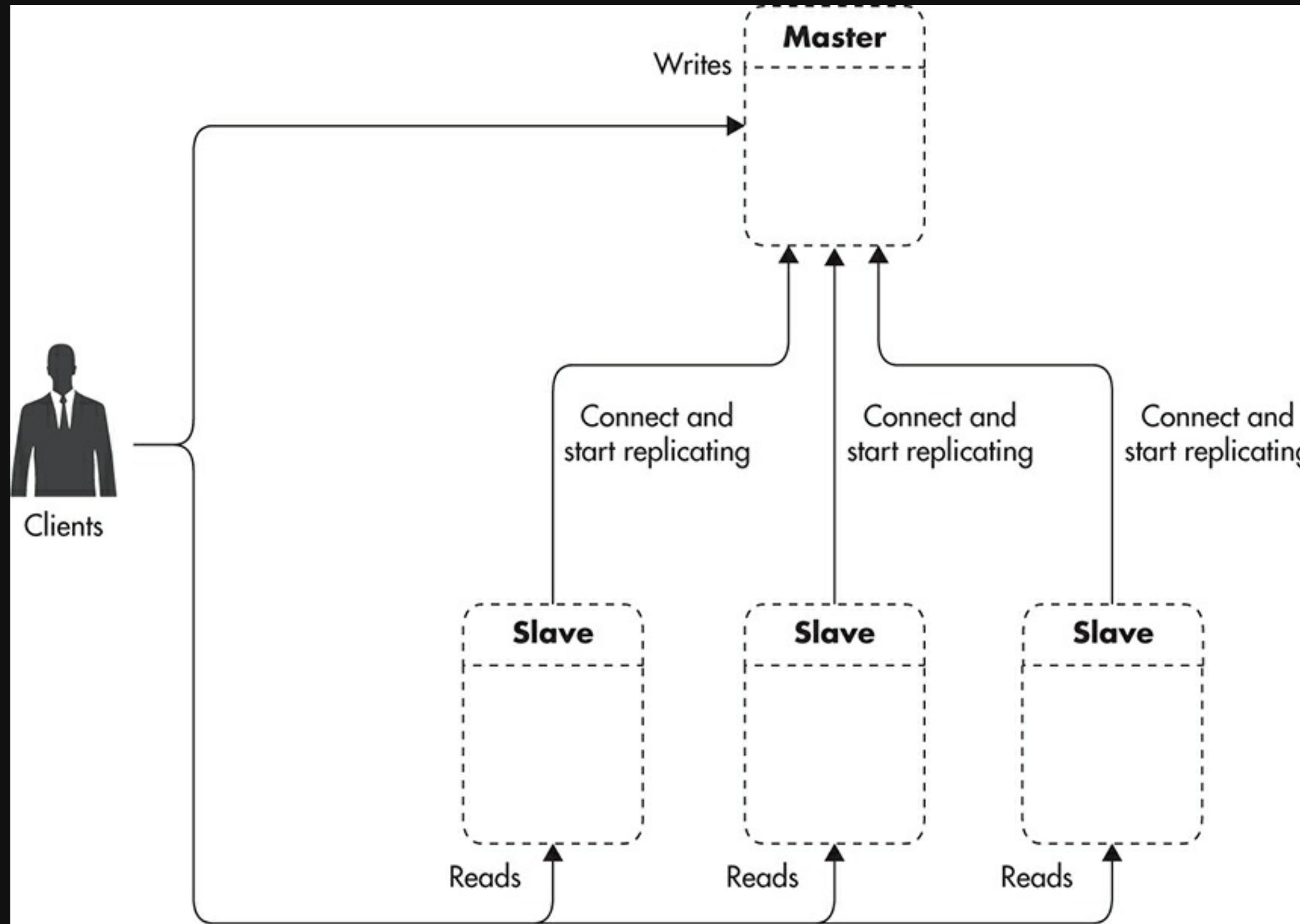
What next!?

Ploygot DataBase for different purpose



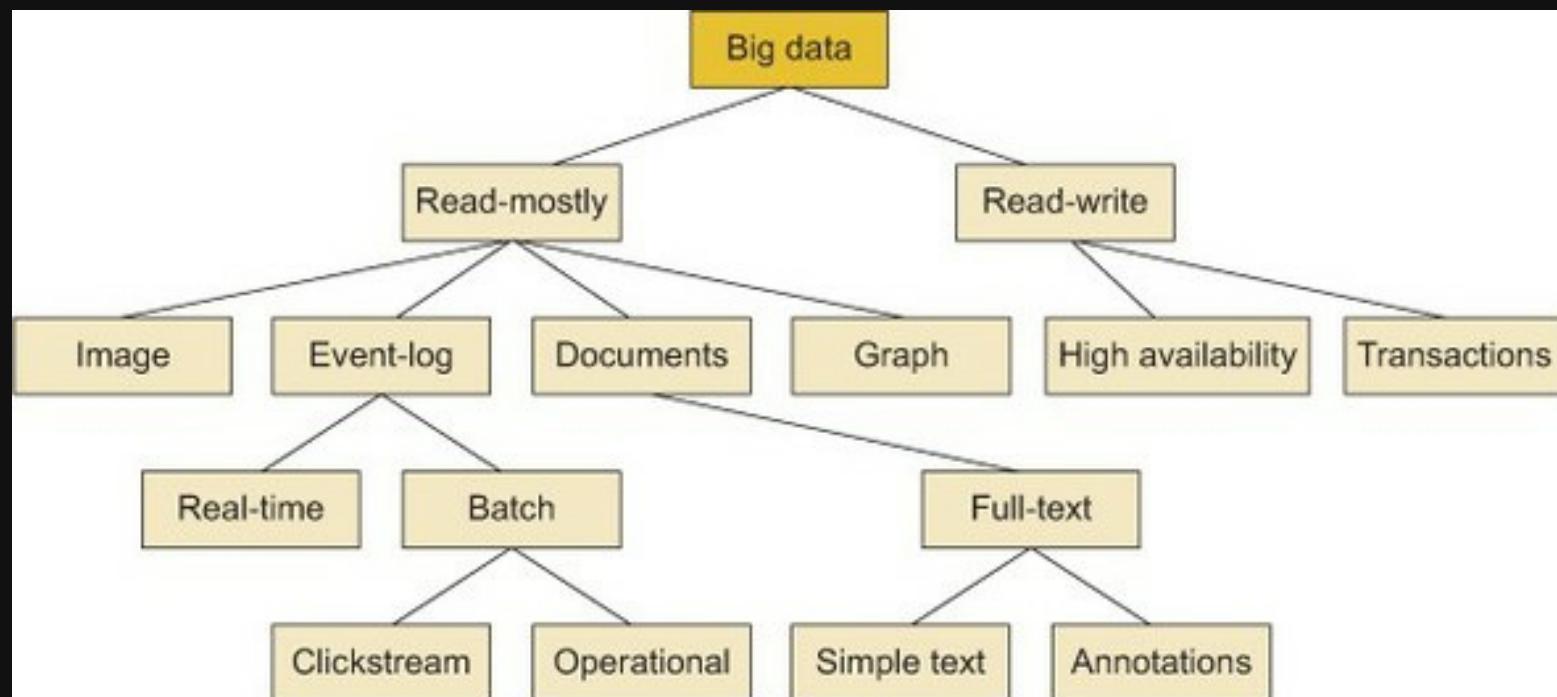
What next!?

Ploygot DataBase for different purpose



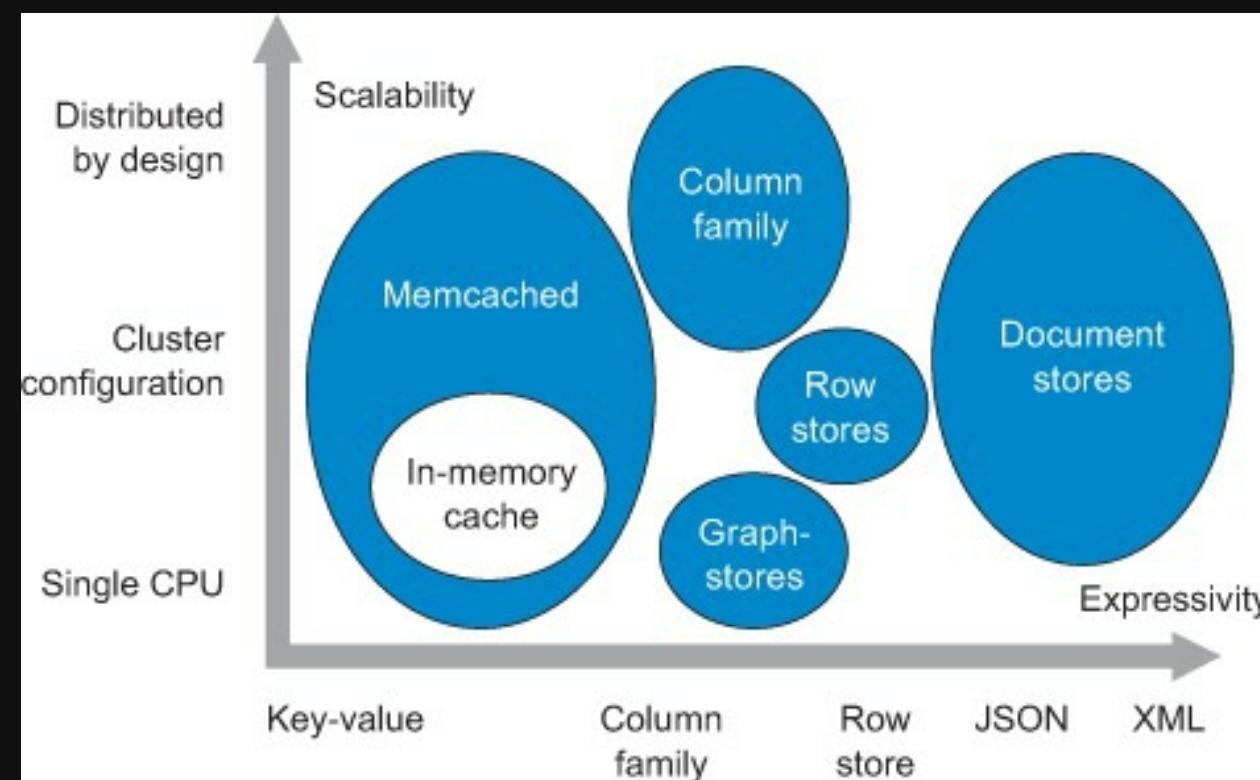
What next!?

Which Technology to choose in DB Layer?
Choose the right data base based on the use case
for data generation

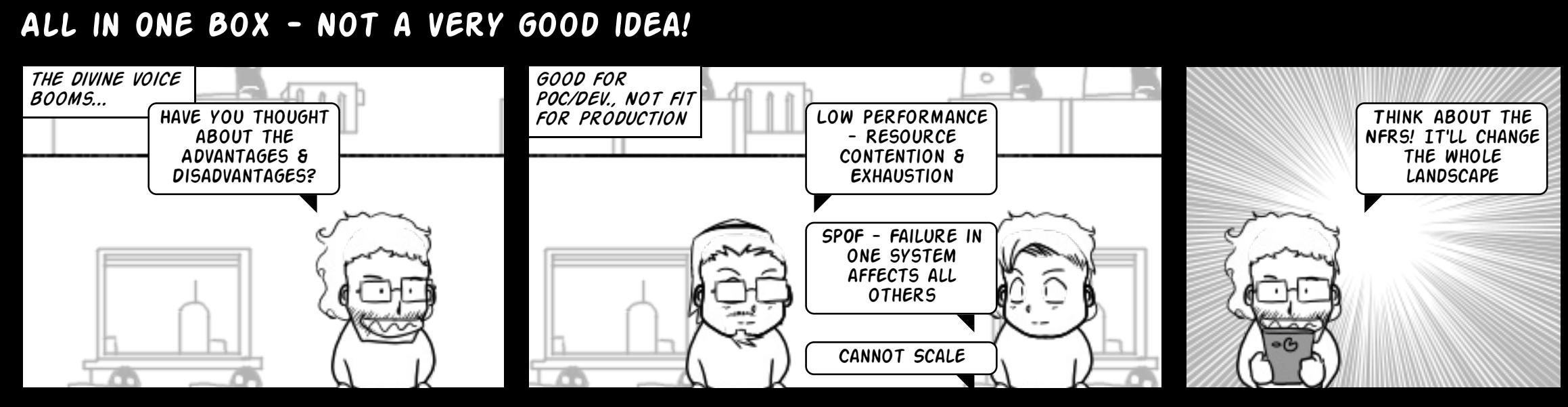


What next!?

Which Technology to choose for nature of data
Choose the right data base based on the type of data

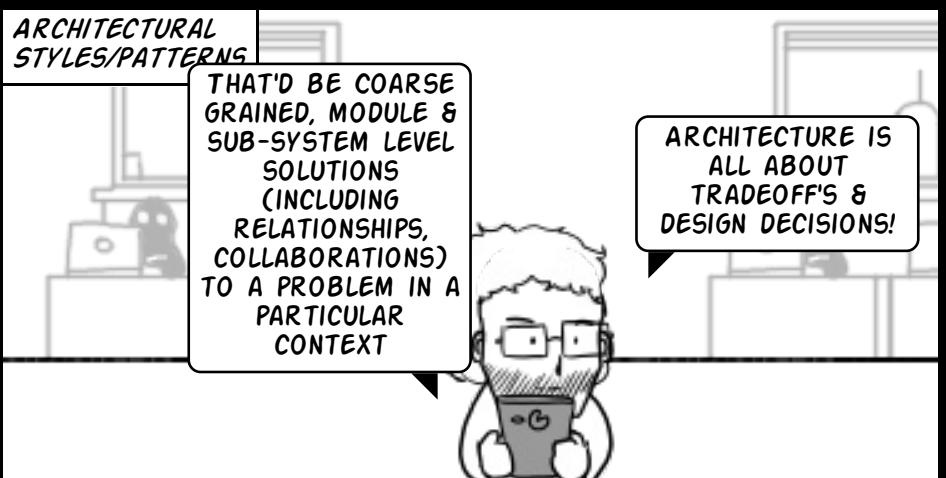
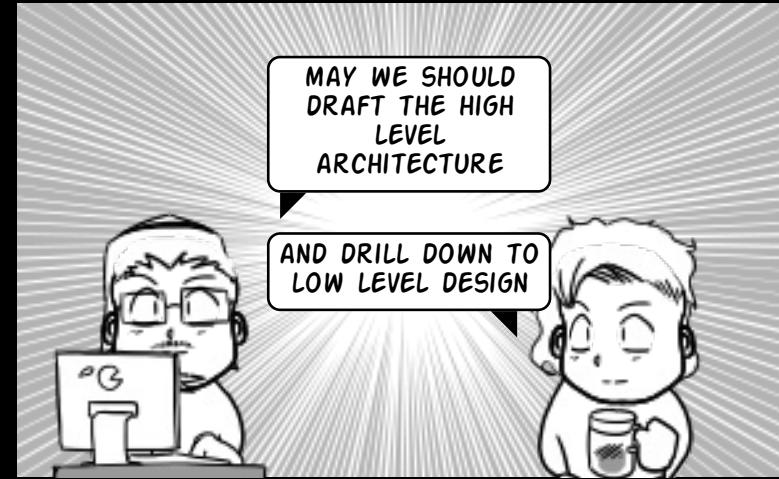


Architecture & Design ...



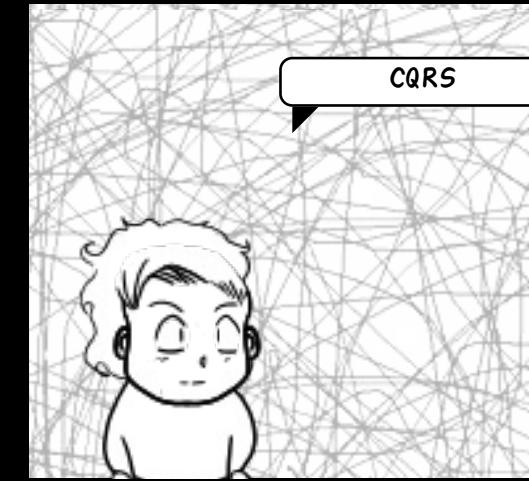
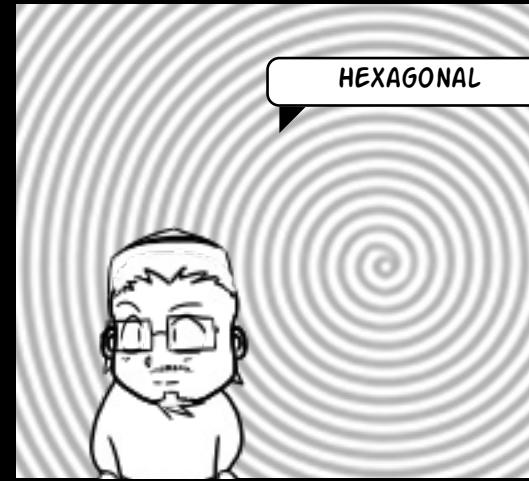
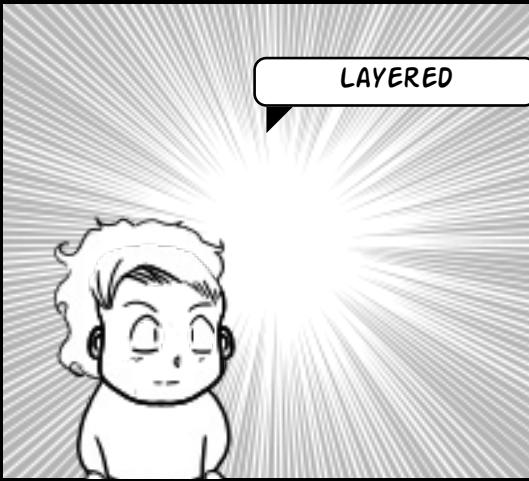
Architecture & Design ...

DEVS THINKING...



Architecture & Design ...

ARCHITECTURAL STYLES/PATTERNS & SELECTION

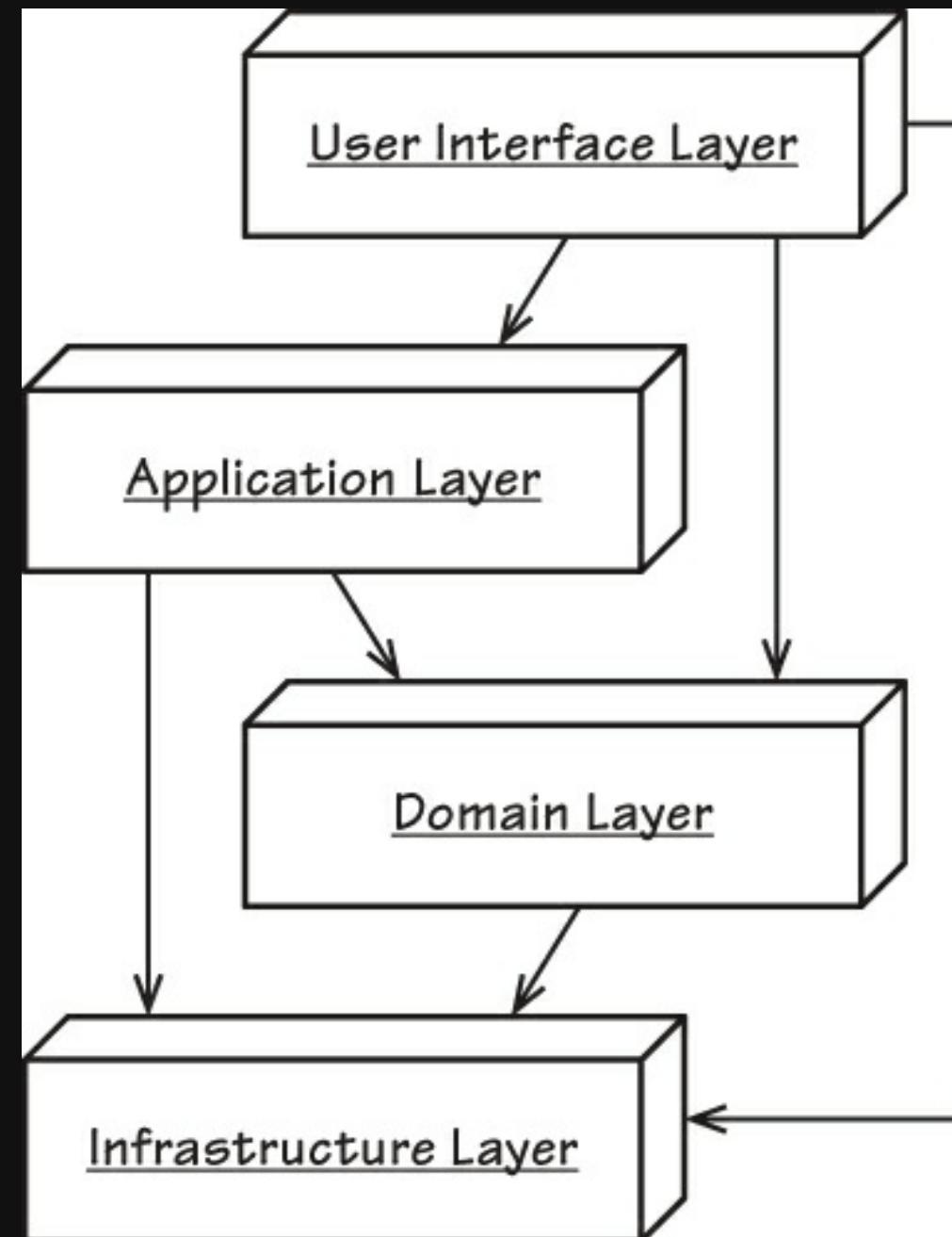


Architecture & Design ...

- * [Layered Architectures](<http://www.softwarearchitectures.com/qa.html>)
- * [Hexagonal Architecture](https://en.wikipedia.org/wiki/Multitier_architecture)
- * [CQRS](https://en.wikipedia.org/wiki/Multitier_architecture)
- * [N-Tier](https://en.wikipedia.org/wiki/Multitier_architecture)
- * [Event Driven / Event Sourcing](https://en.wikipedia.org/wiki/Multitier_architecture)

Architecture & Design ...

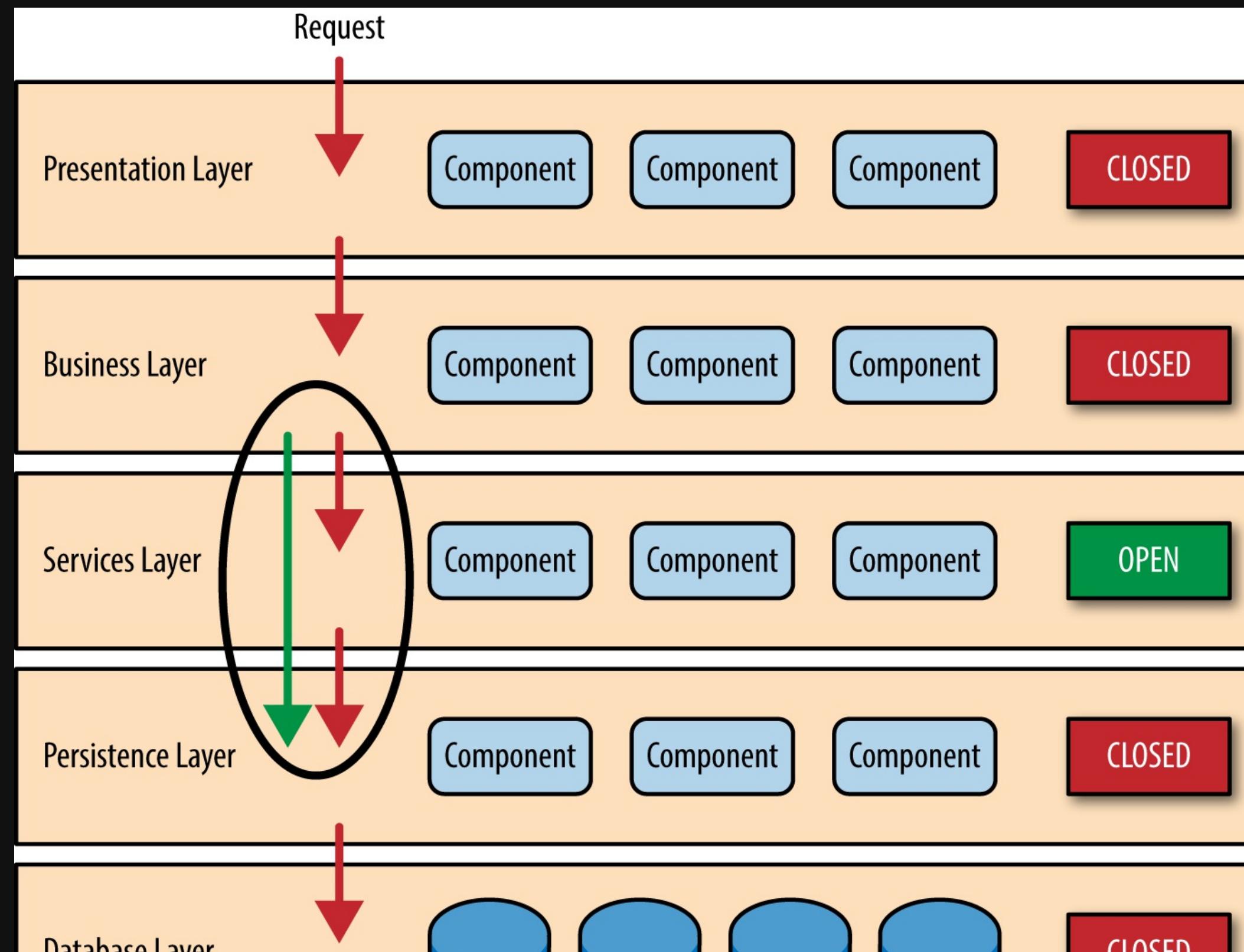
Layered Architecture



Separation of Concerns - Simple Logical
Decomposition, Top Down with Isolation

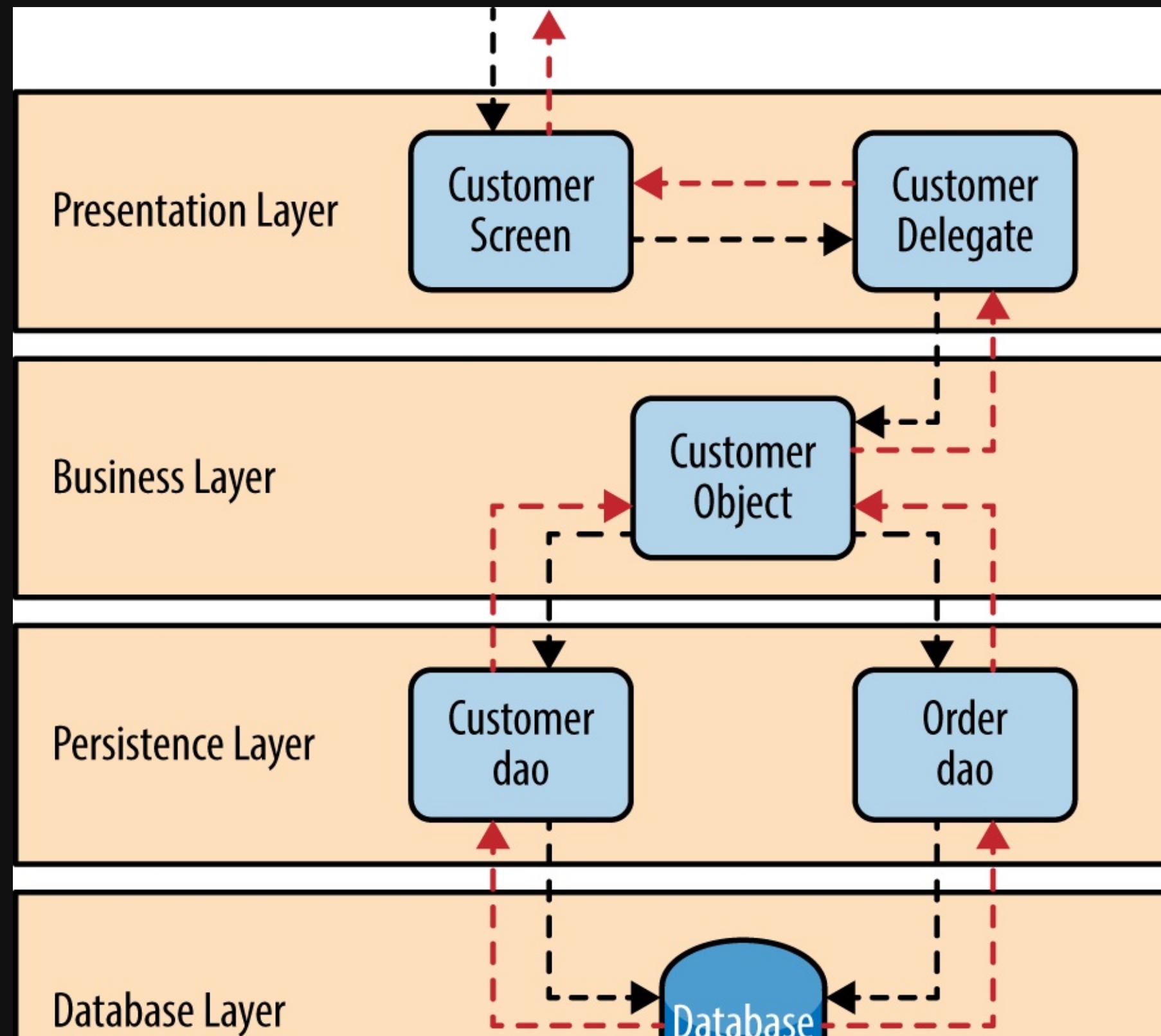
Architecture & Design ...

Layered Architecture



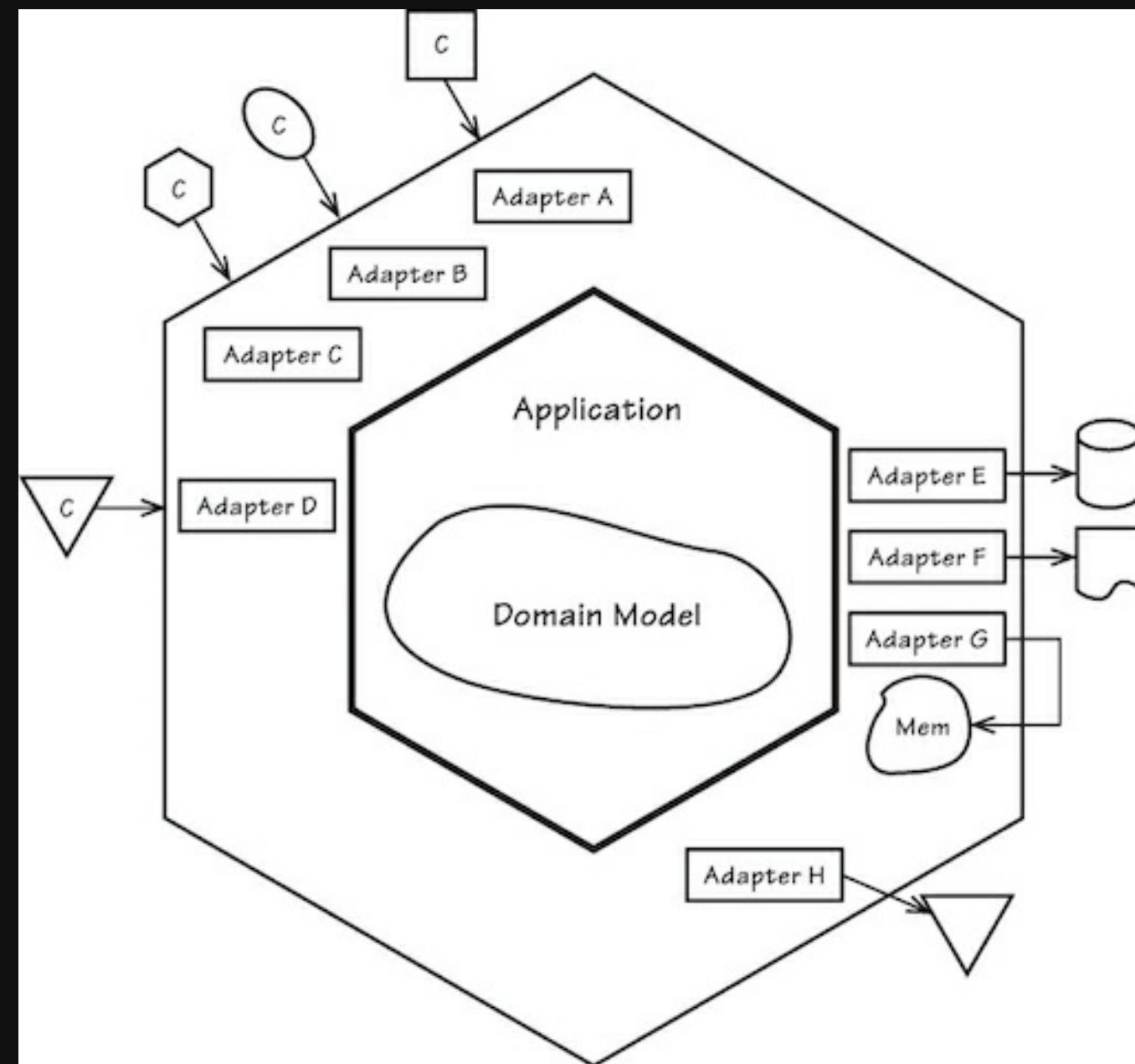
Architecture & Design ...

Layered Architecture



Architecture & Design ...

Hexagonal Architecture



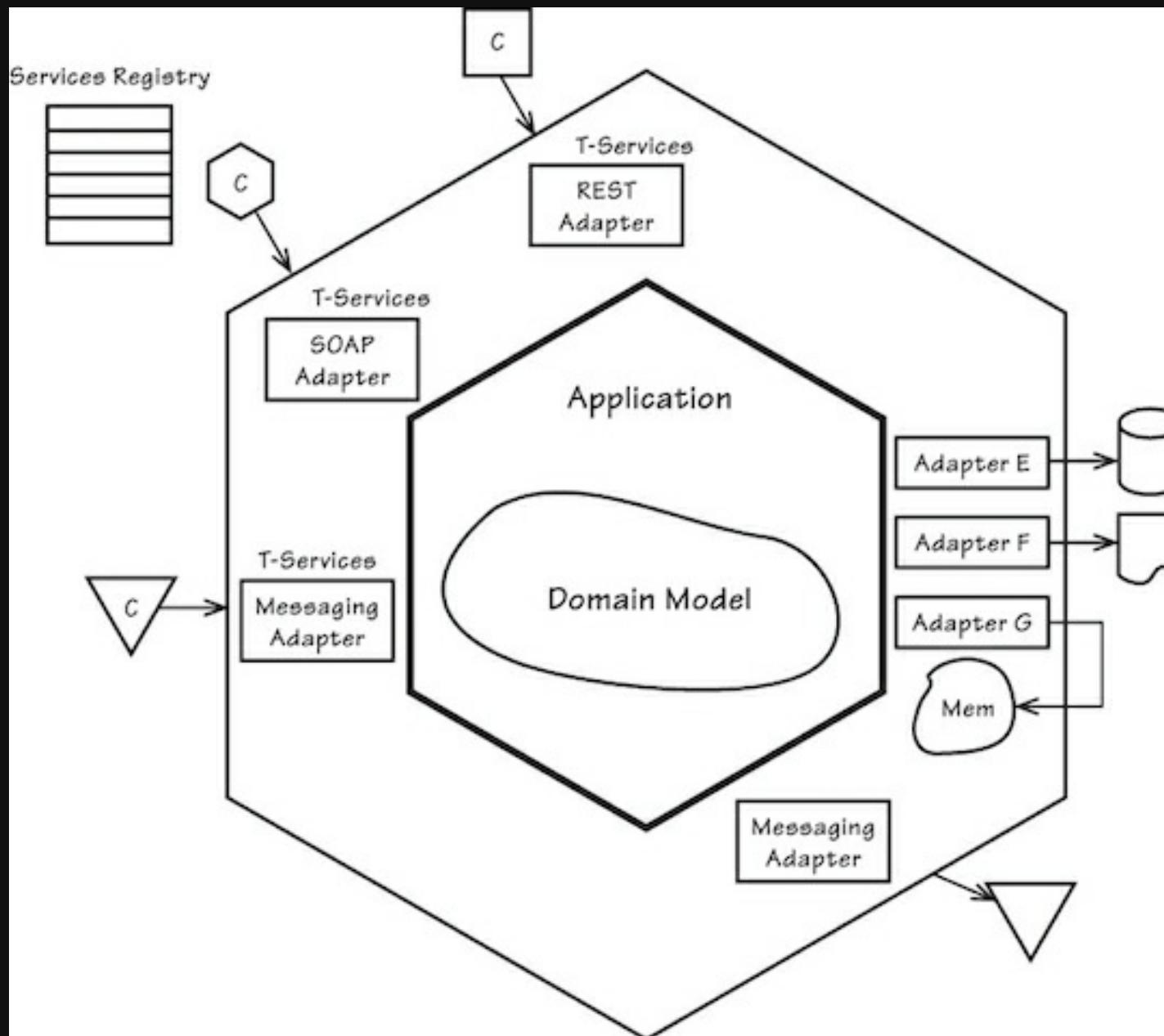
Architecture & Design ...

Hexagonal Architecture

Allow an application to equally be driven by users, programs, automated test or batch scripts, and to be developed and tested in isolation from its eventual run-time devices and databases.

Architecture & Design ...

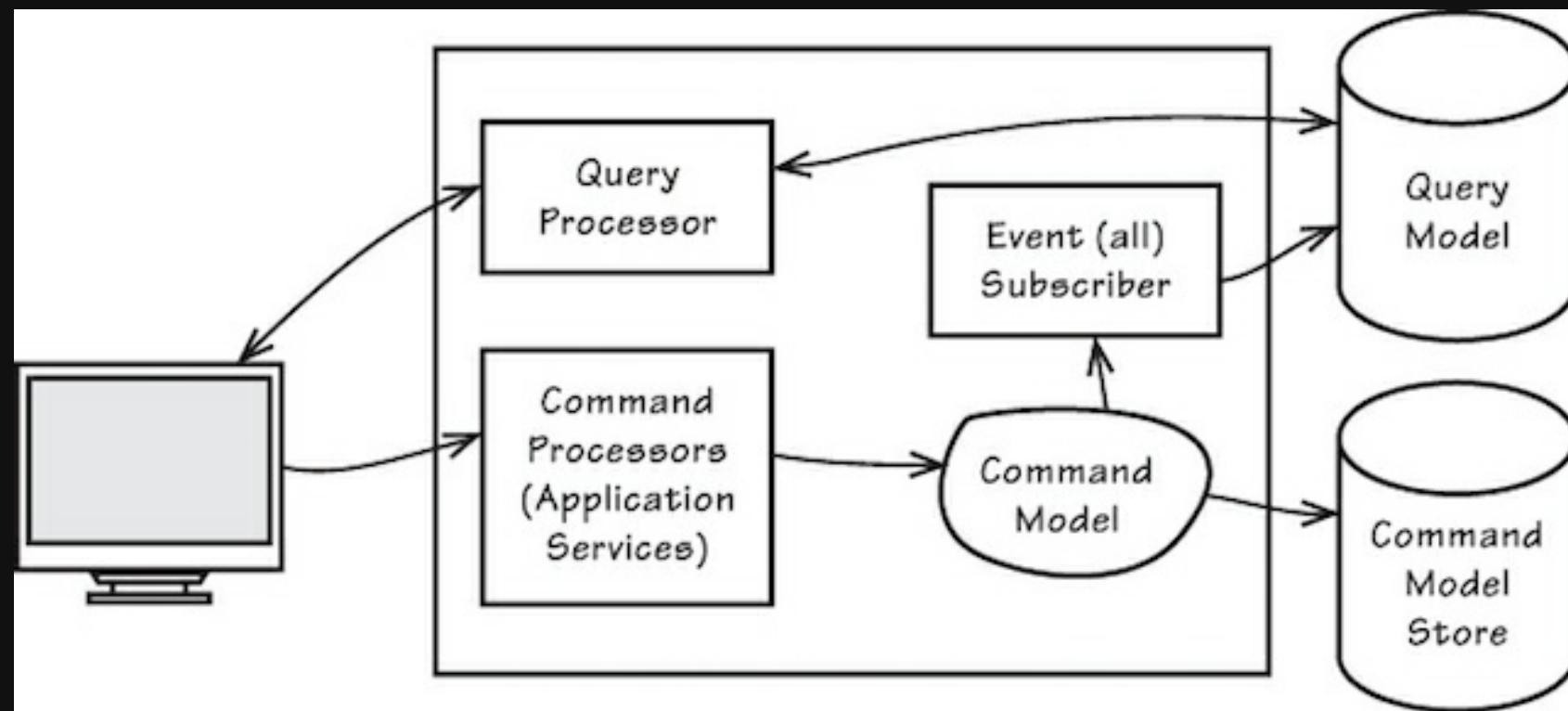
Hexagonal Architecture SOA



The concepts of Adapters with CoreDomain & with SOA

Architecture & Design ...

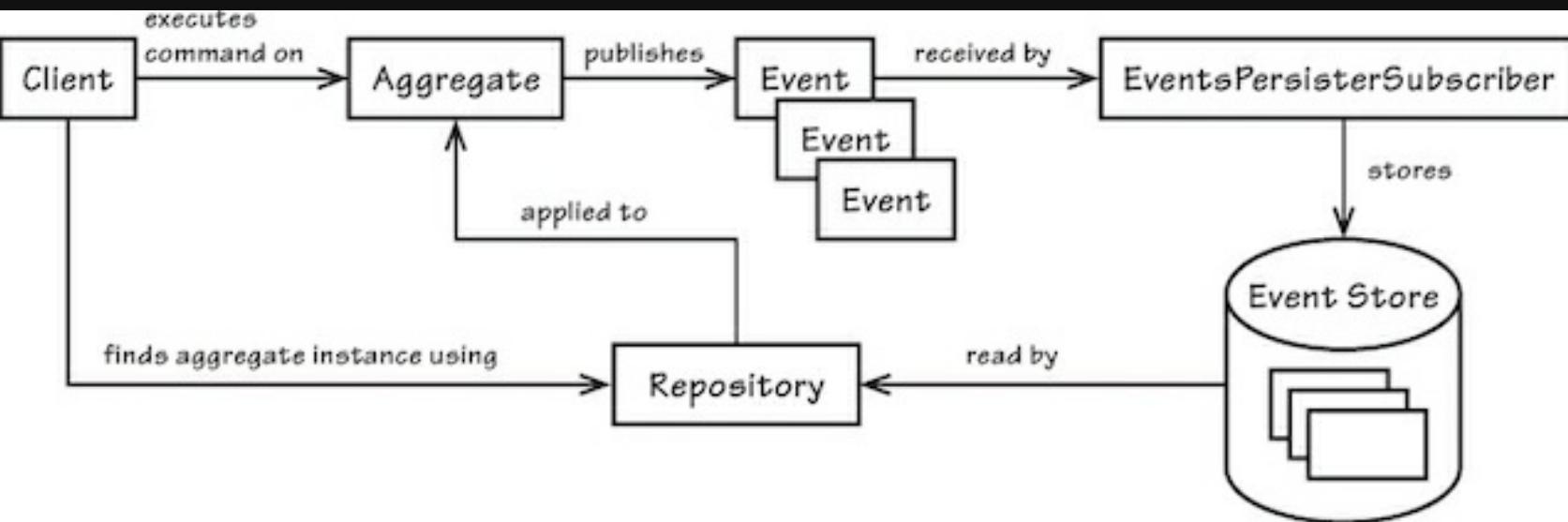
CQRS



Separate Flow for Queries (Reads) and Create (Writes)

Architecture & Design ...

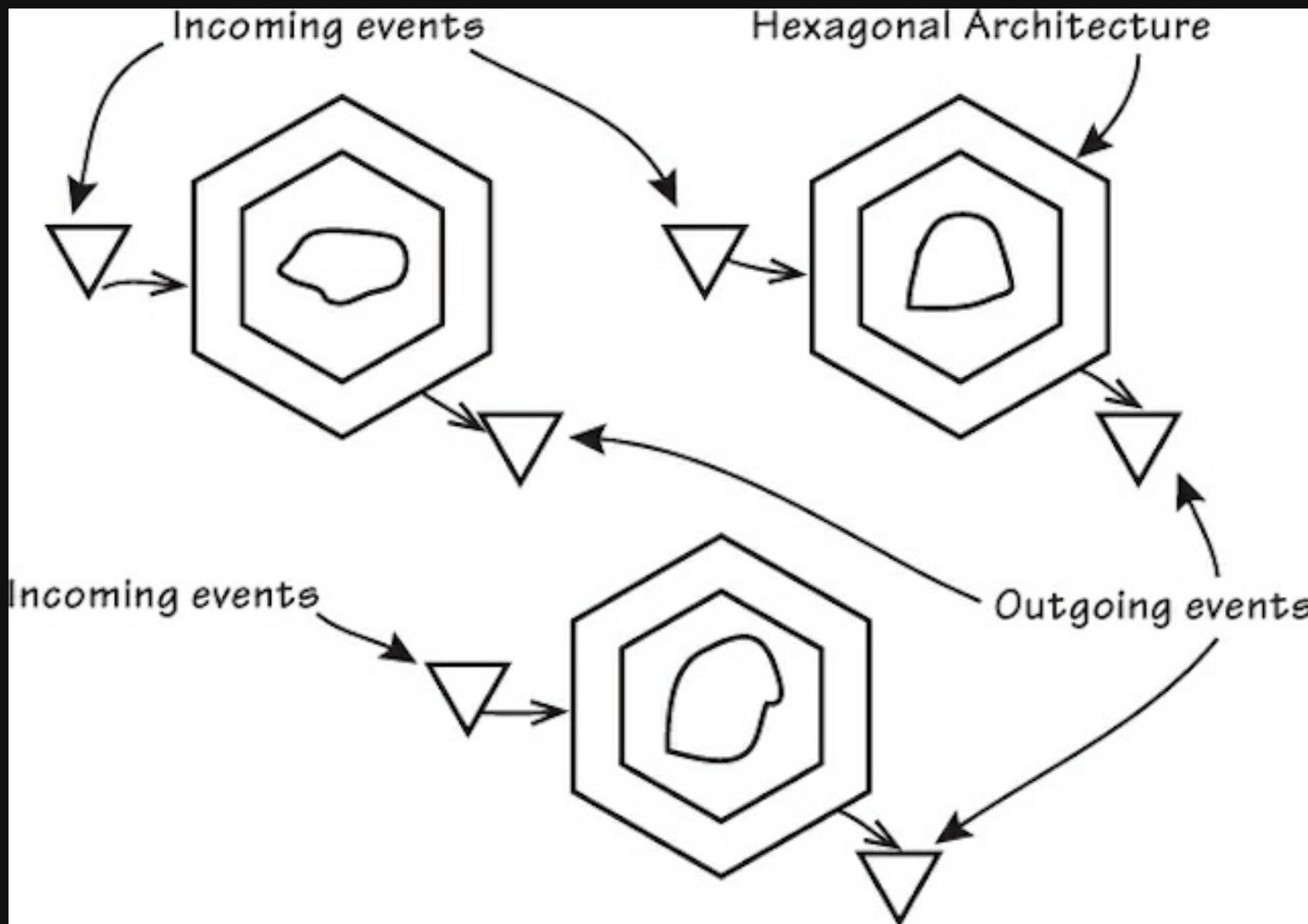
Event Sourcing



Events are Immutable! There are no Updates & Deletes. Changes in state are stored as new events.

Architecture & Design ...

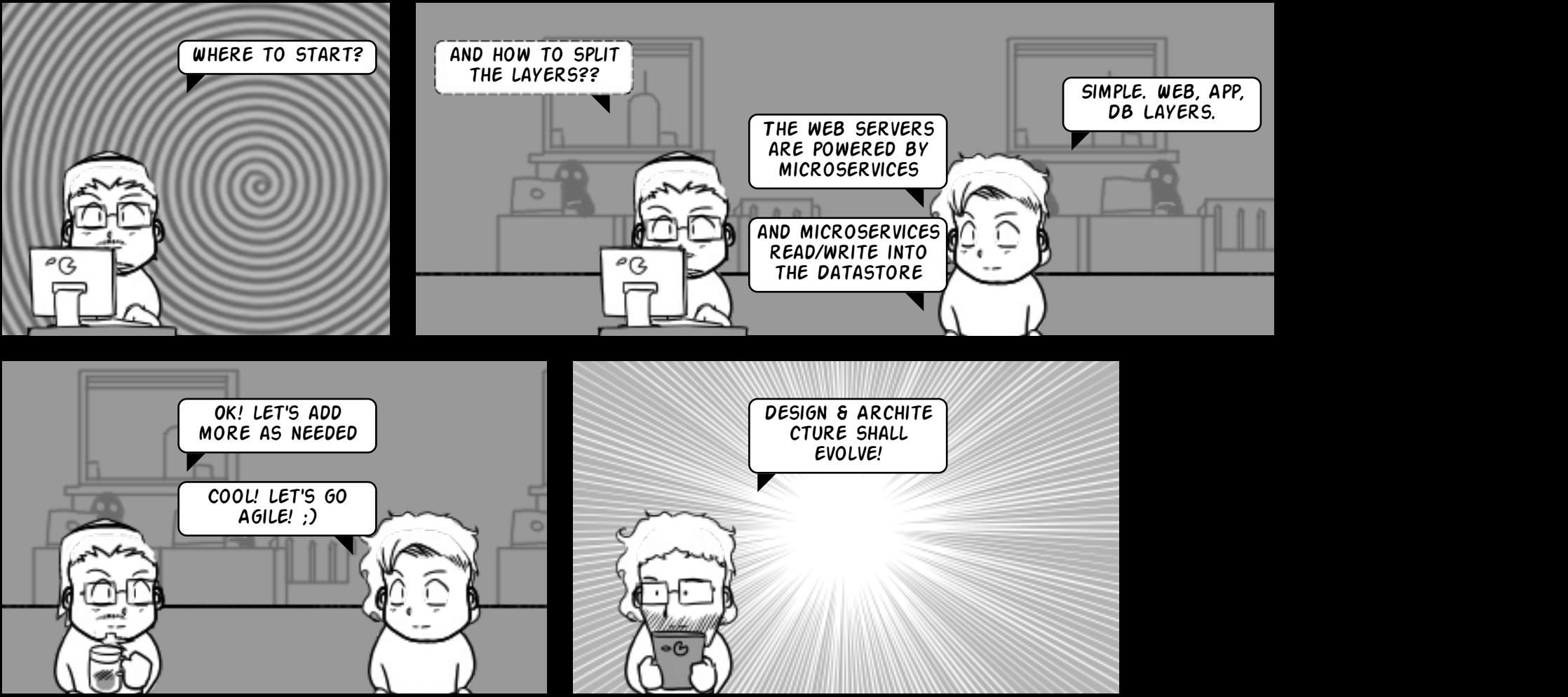
Event Driven



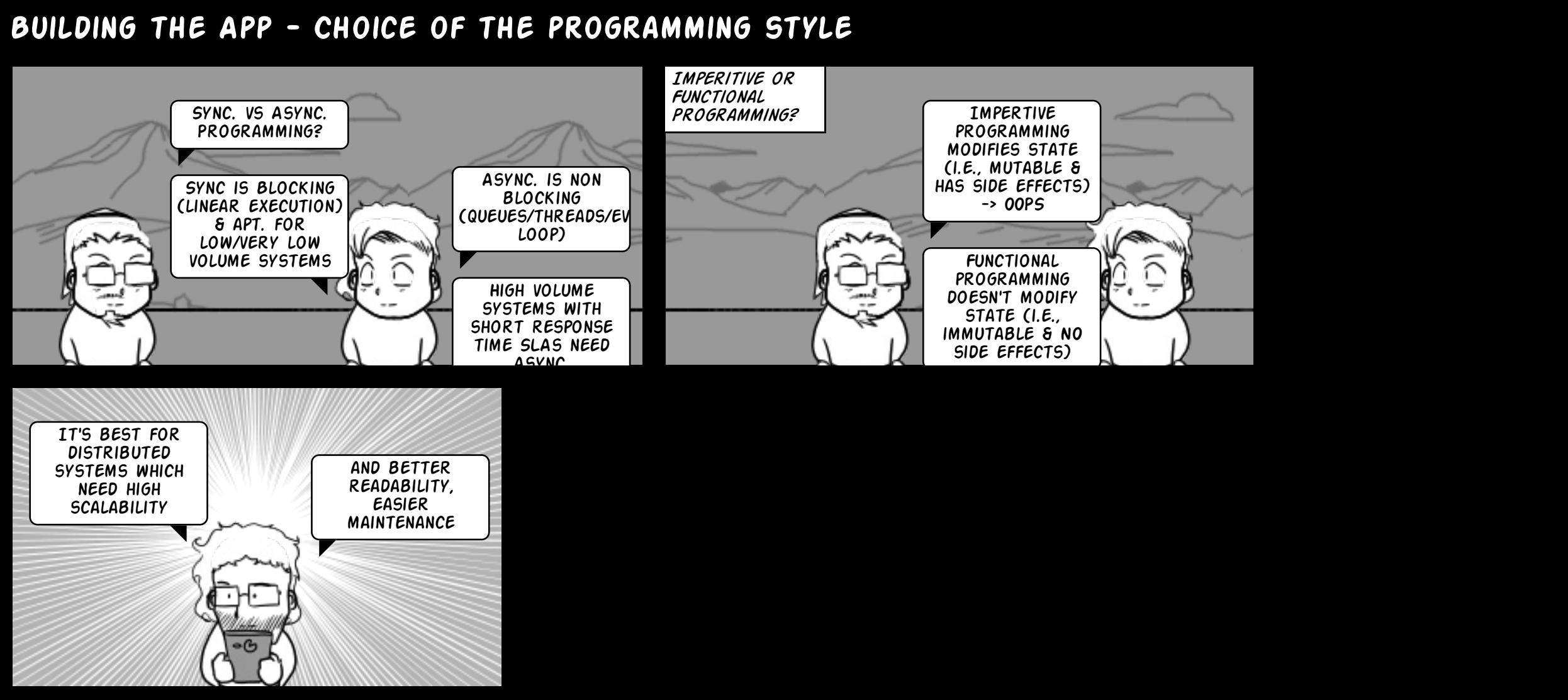
Different System integrate using Domain Events

Architecture & Design ...

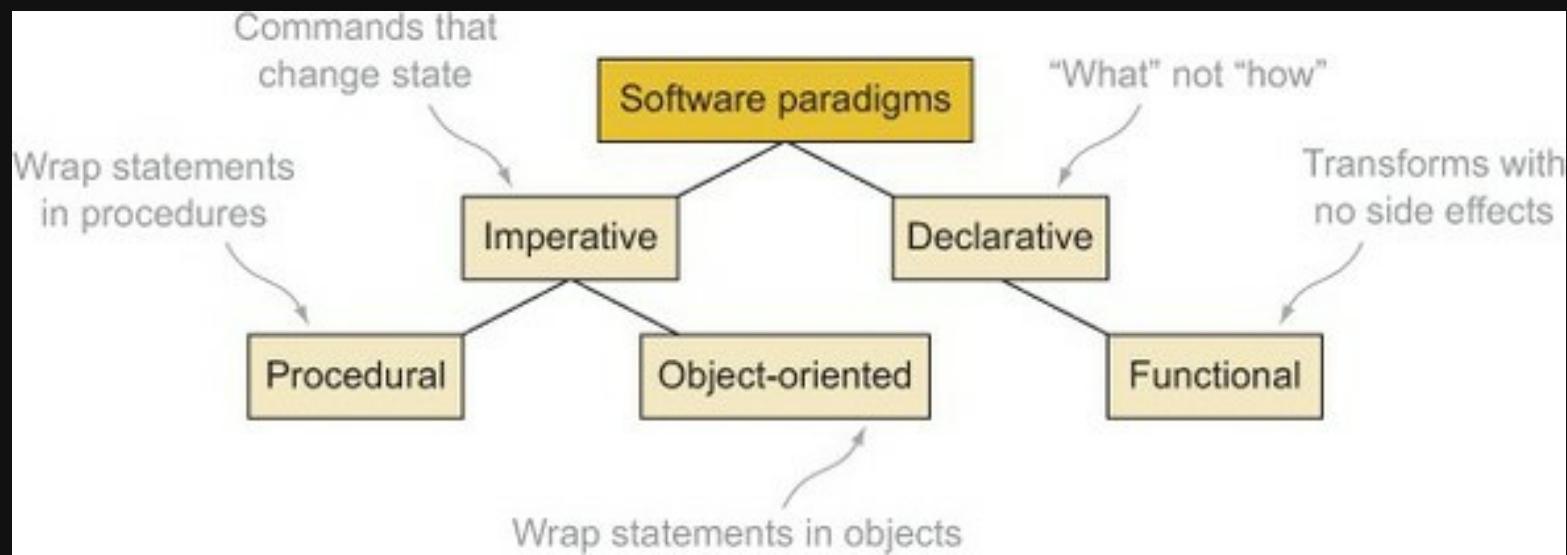
N-TIER ARCHITECTURE & LAYERS



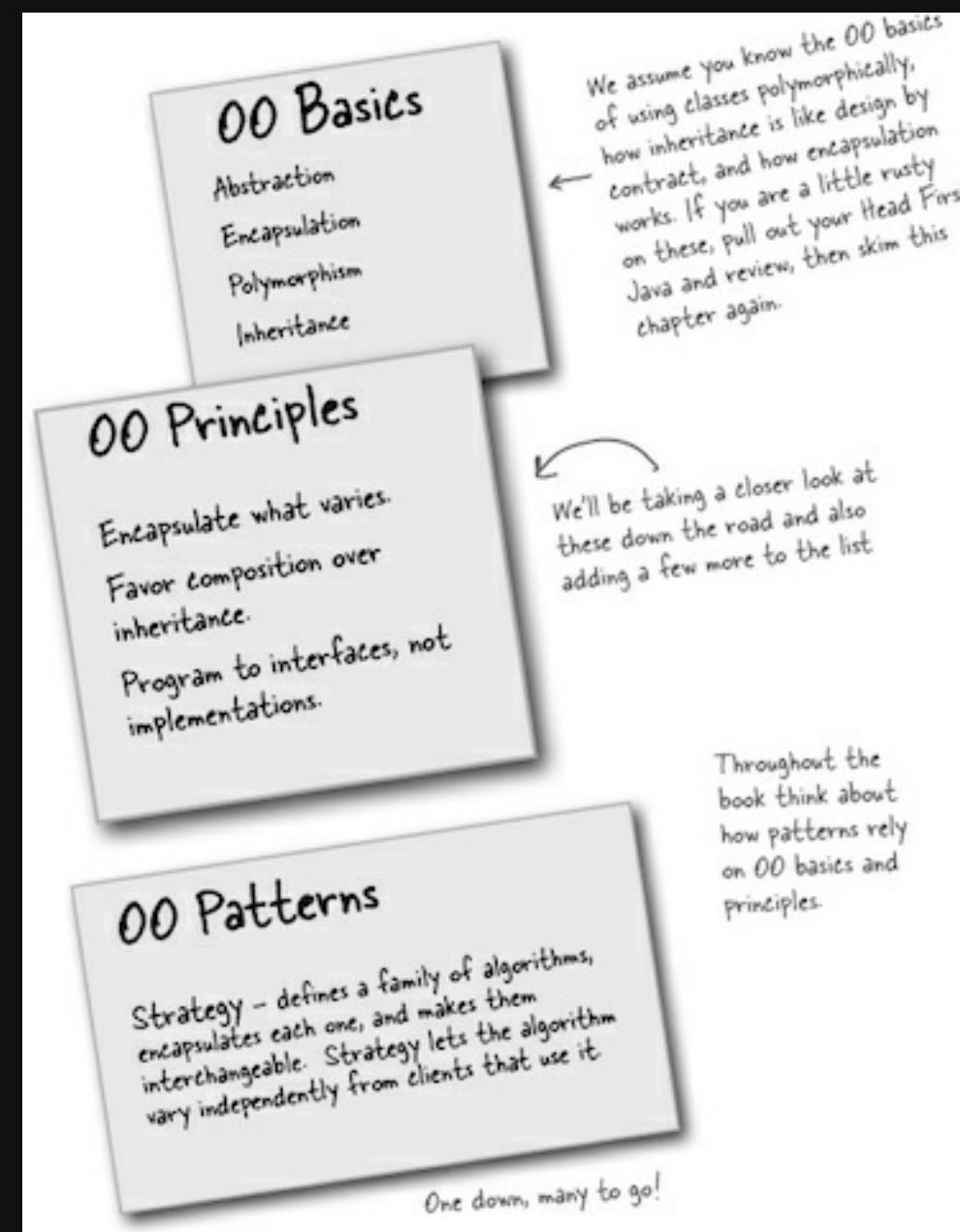
Architecture & Design ...



Architecture & Design ...

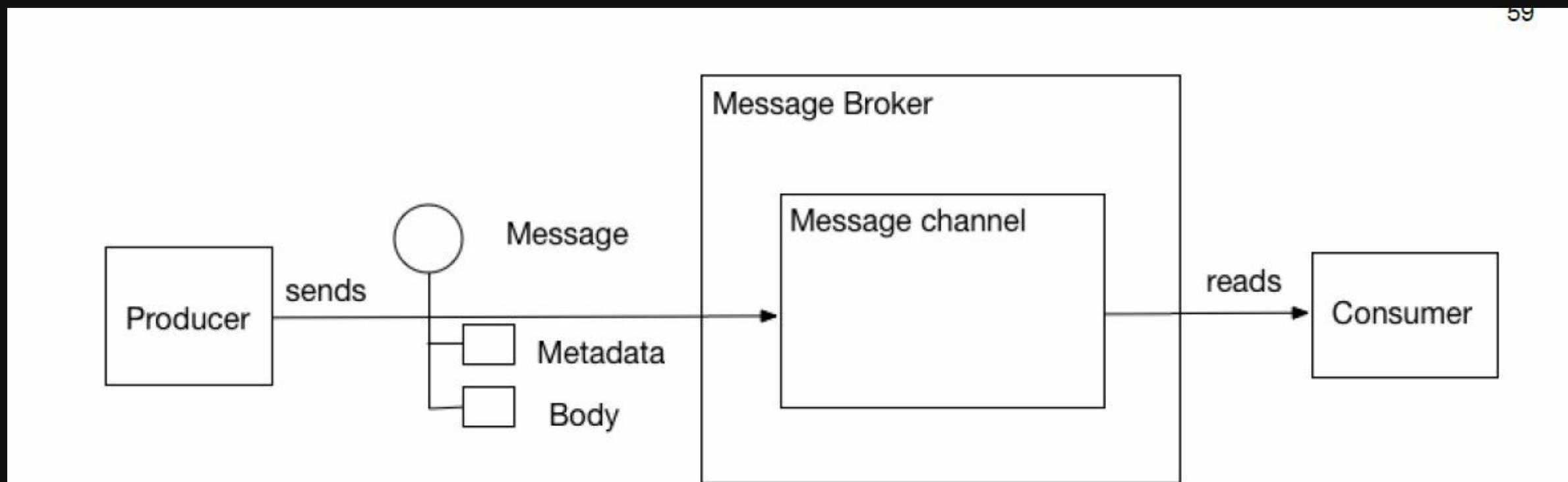


Architecture & Design ...



Architecture & Design ...

59



Synchronous does not perform all the time

AMQP based Protocol Product Rabbit MQ

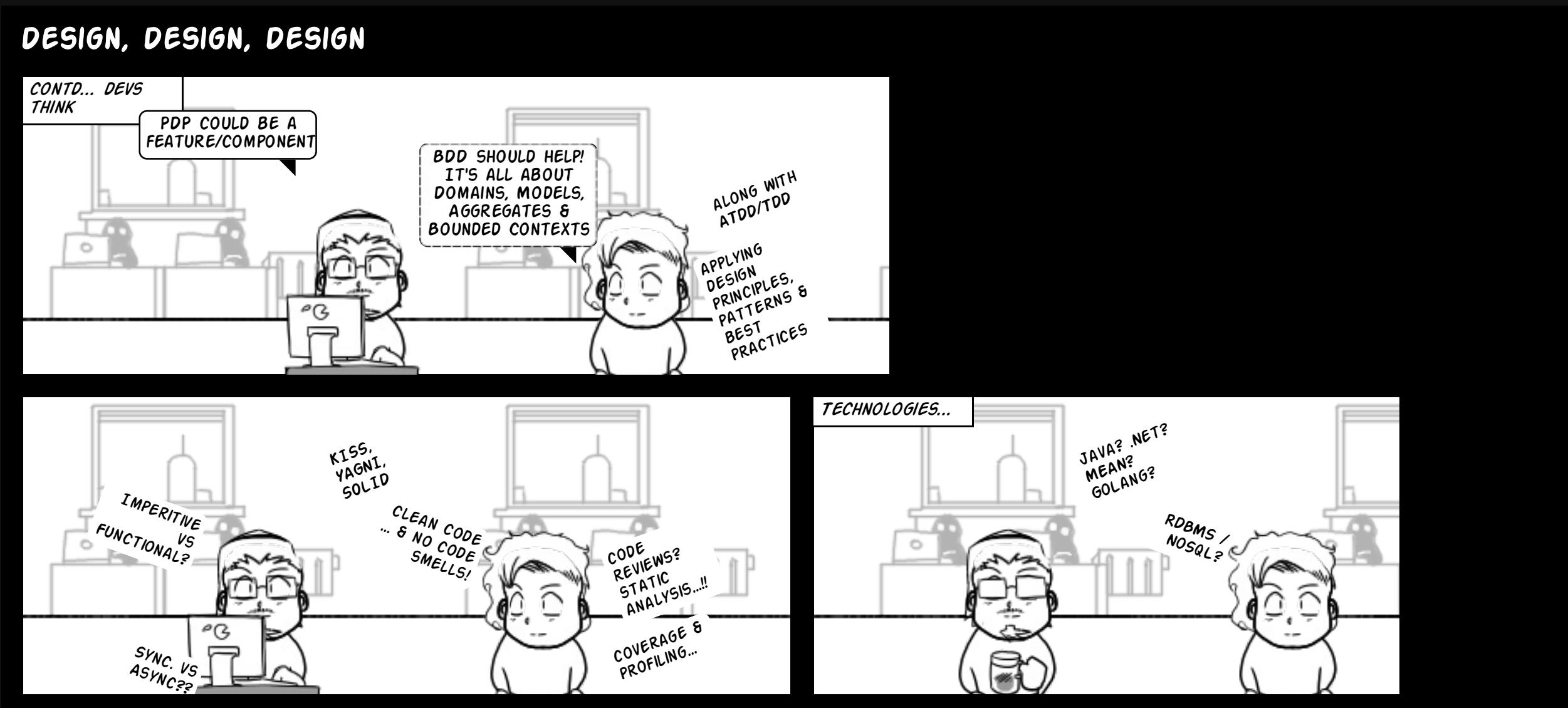
Architecture & Design ...

<https://maryrosecook.com/blog/post/a-practical-introduction-to-functional-programming>

https://blog.heroku.com/concurrency_is_not_parallelism

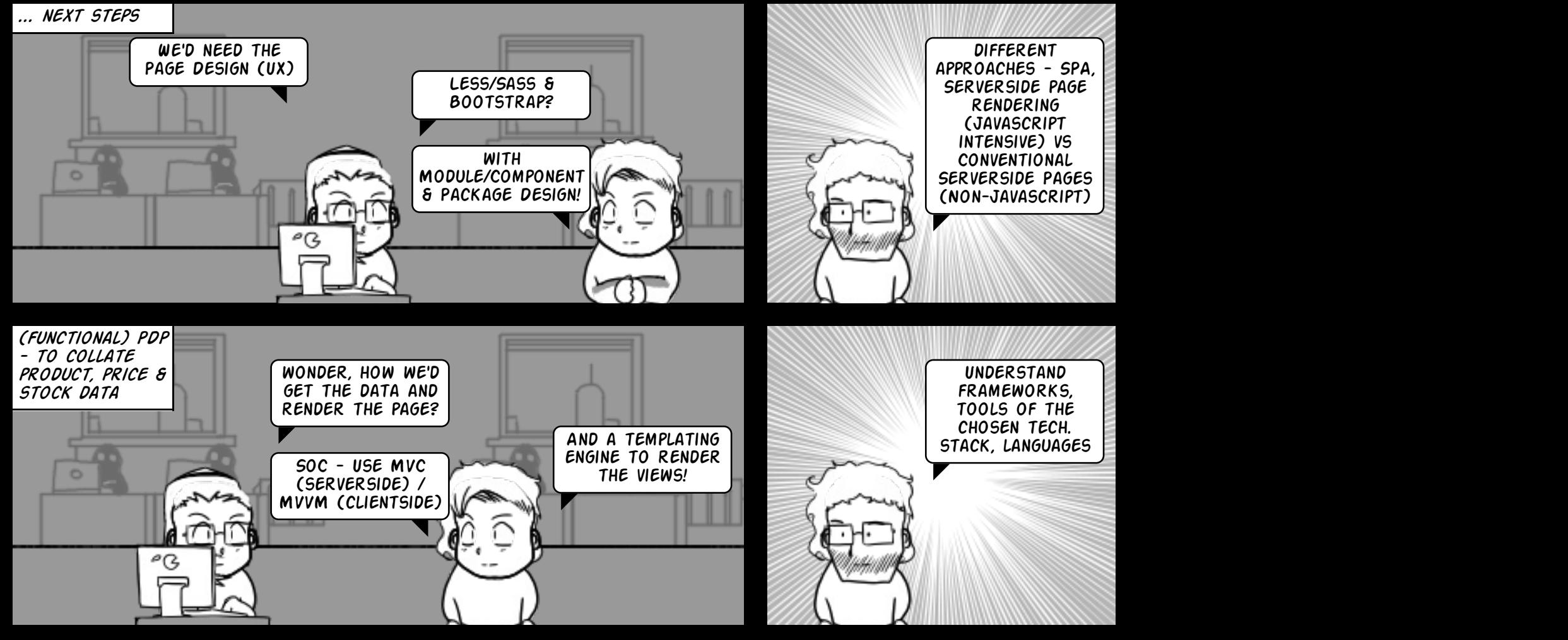
<https://talks.golang.org/2012/waza.slide#1>

Architecture & Design ...



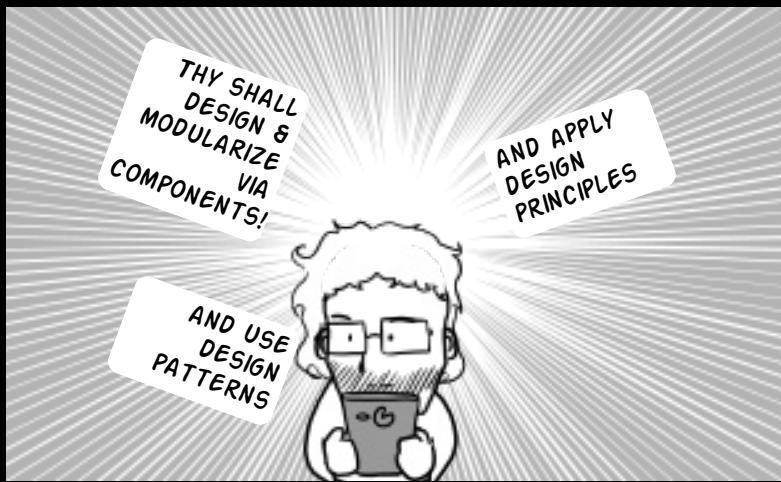
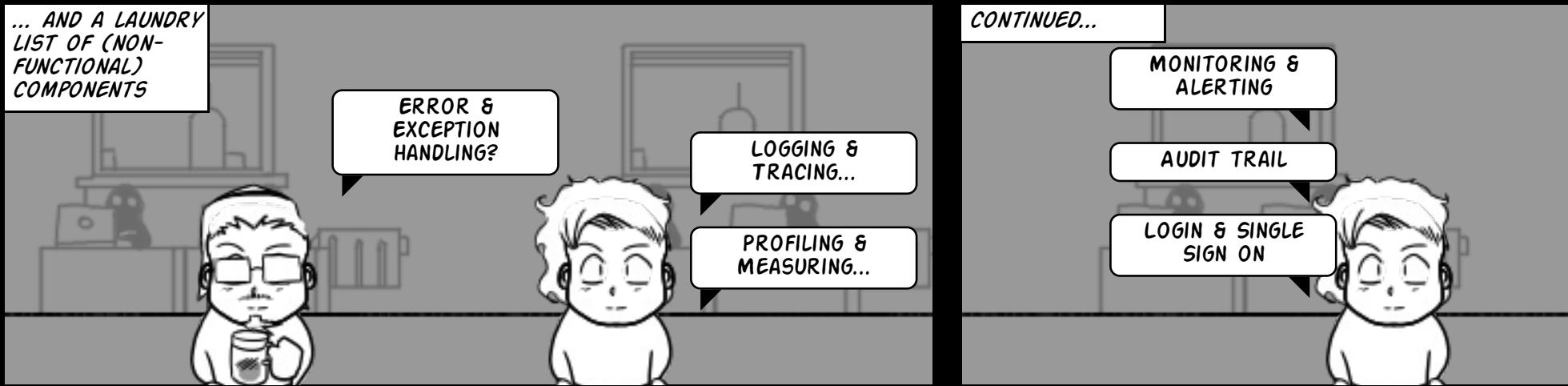
Architecture & Design ...

BREAKING DOWN INTO MANAGABLE PIECES - FUNCTIONAL COMPONENTS

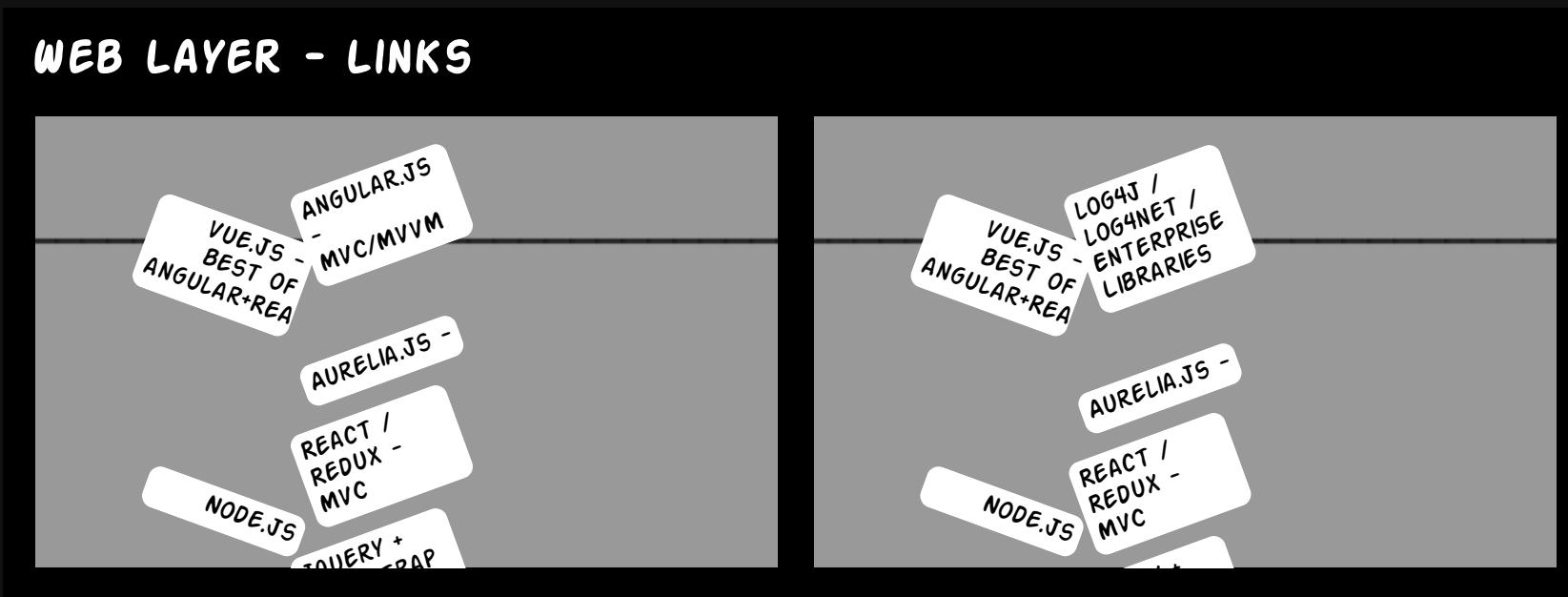


Architecture & Design ...

BREAKING DOWN INTO MANAGABLE PIECES - NON-FUNCTIONAL COMPONENTS



Architecture & Design ...



Architecture & Design ...

<https://msdn.microsoft.com/en-us/library/ms998530.aspx>

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=11711>

<https://msdn.microsoft.com/en-us/library/ff921345.aspx>

<https://msdn.microsoft.com/en-gb/library/ff648138.aspx>

<http://dataguidance.codeplex.com/releases>

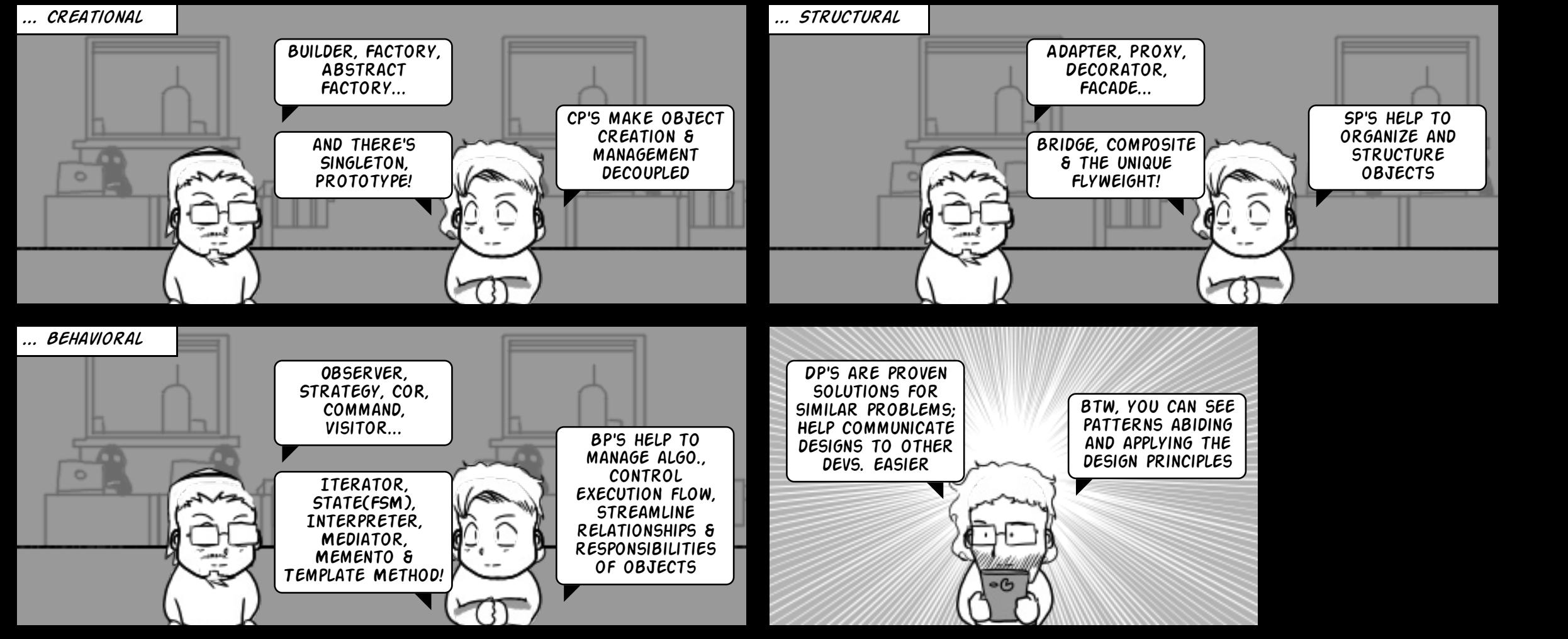
Architecture

<http://shapingsoftware.com/>

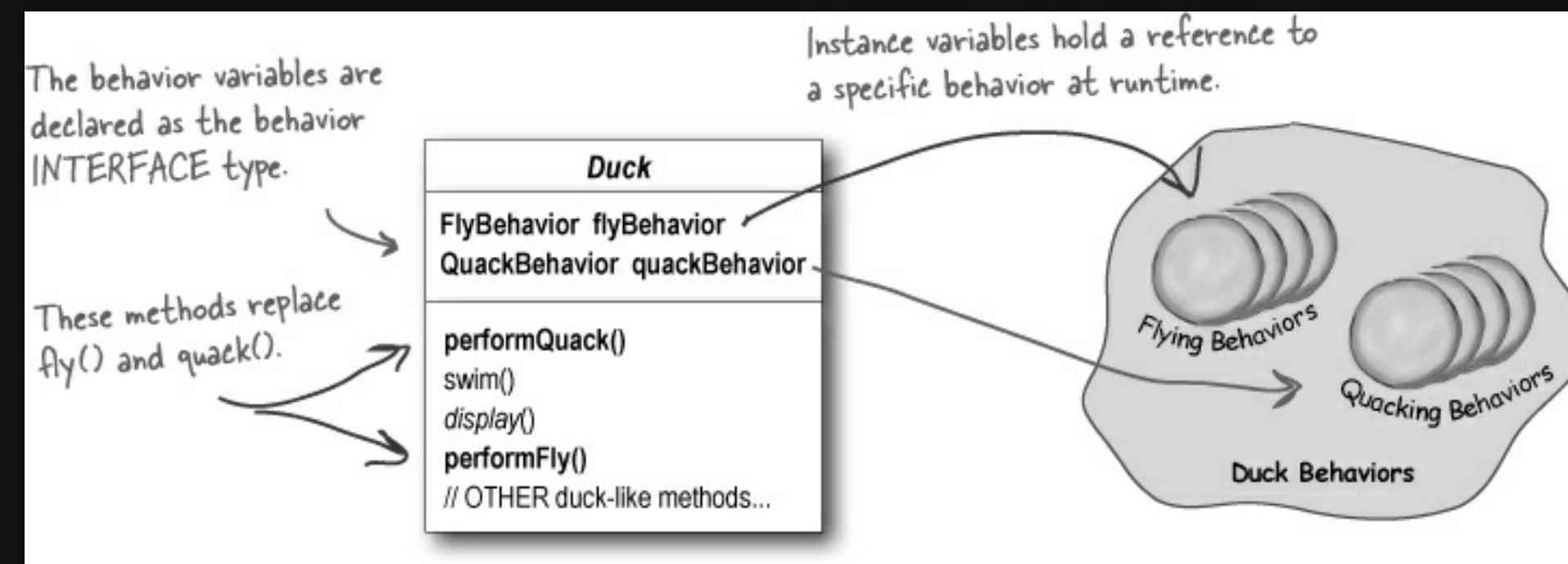
<http://sourcesofinsight.com/>

Architecture & Design ...

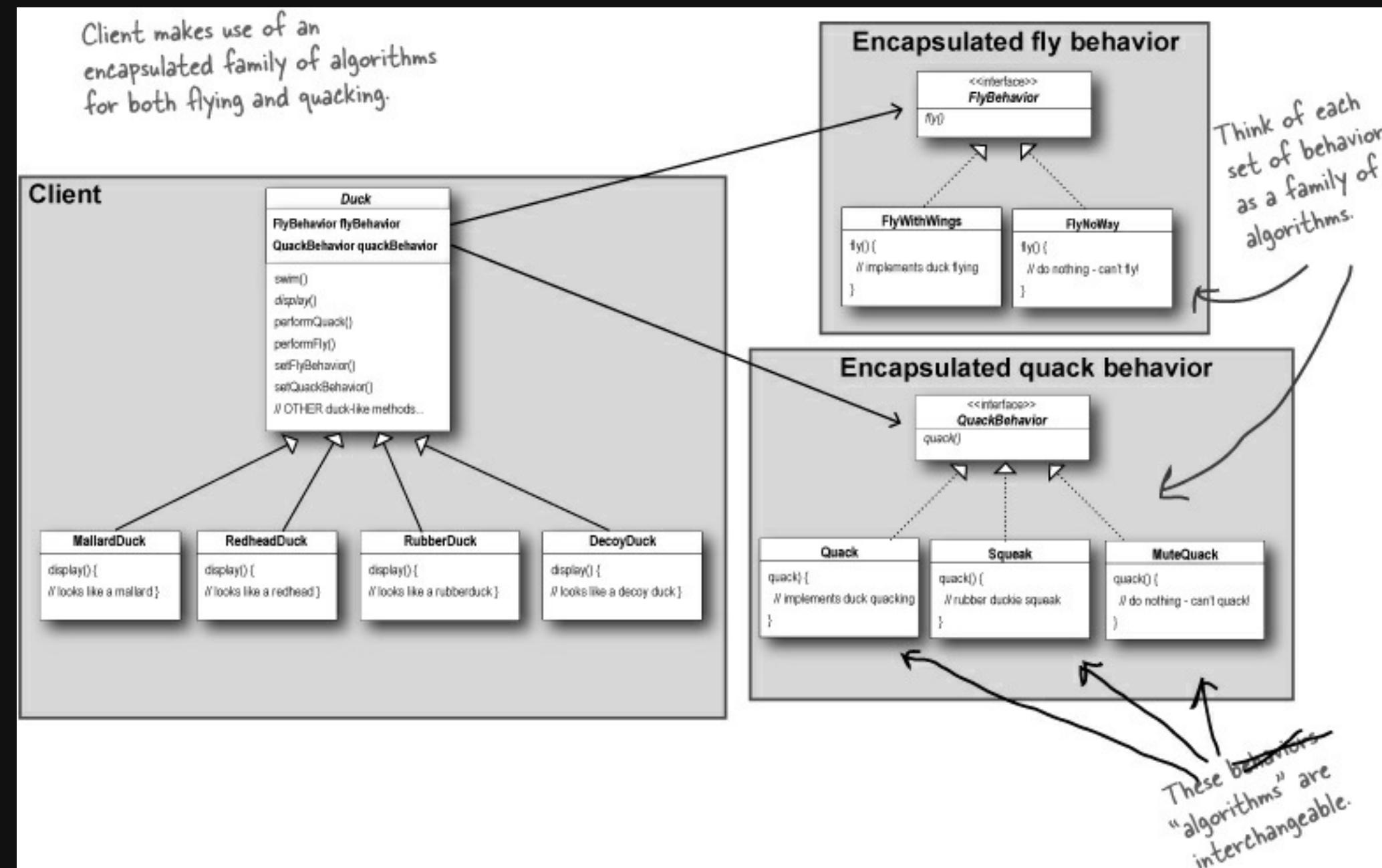
DESIGN PATTERNS - FINE GRAINED, CLASS LEVEL SOLUTION TO A PROBLEM IN A PARTICULAR CONTEXT



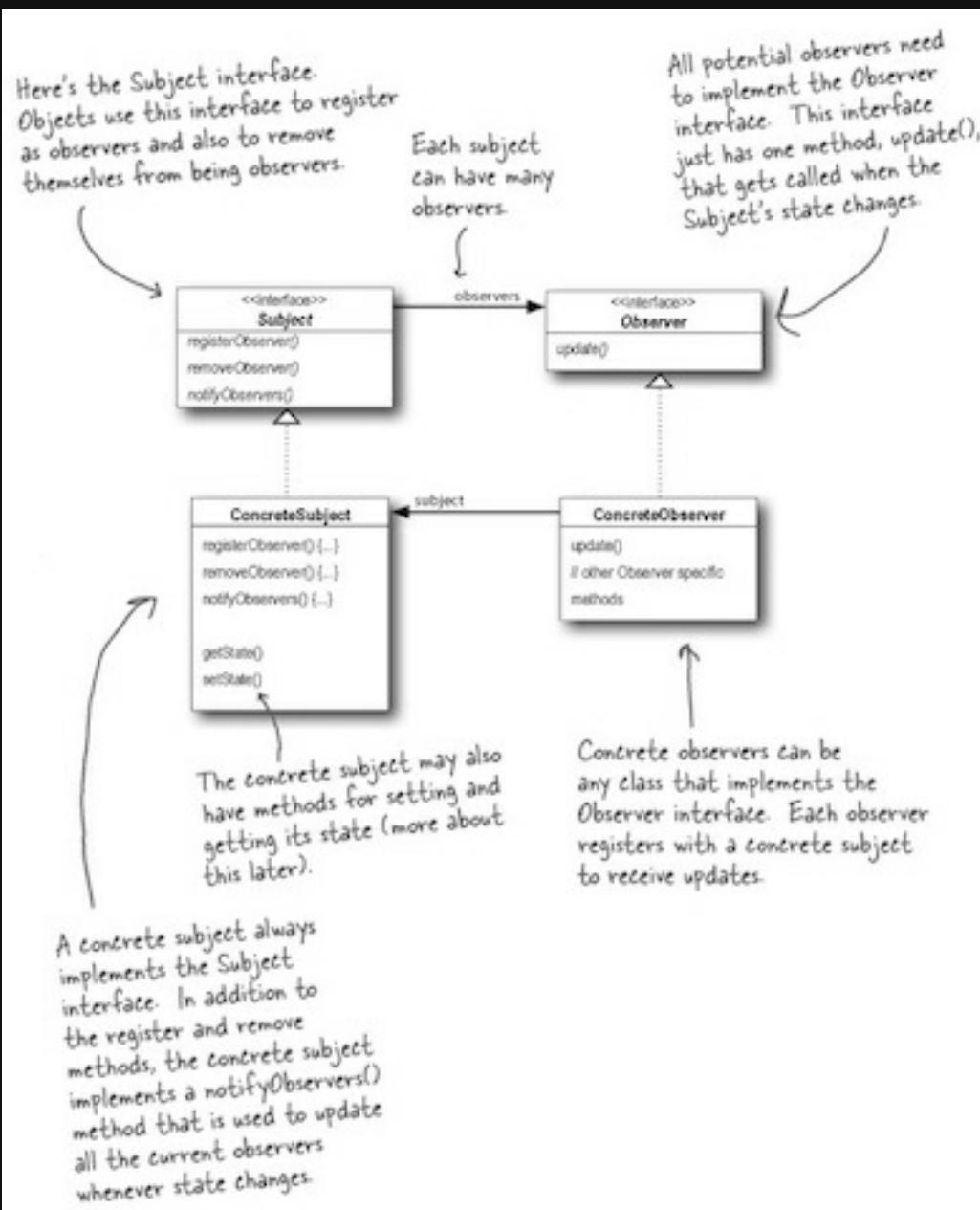
Architecture & Design ...



Architecture & Design ...



Architecture & Design ...



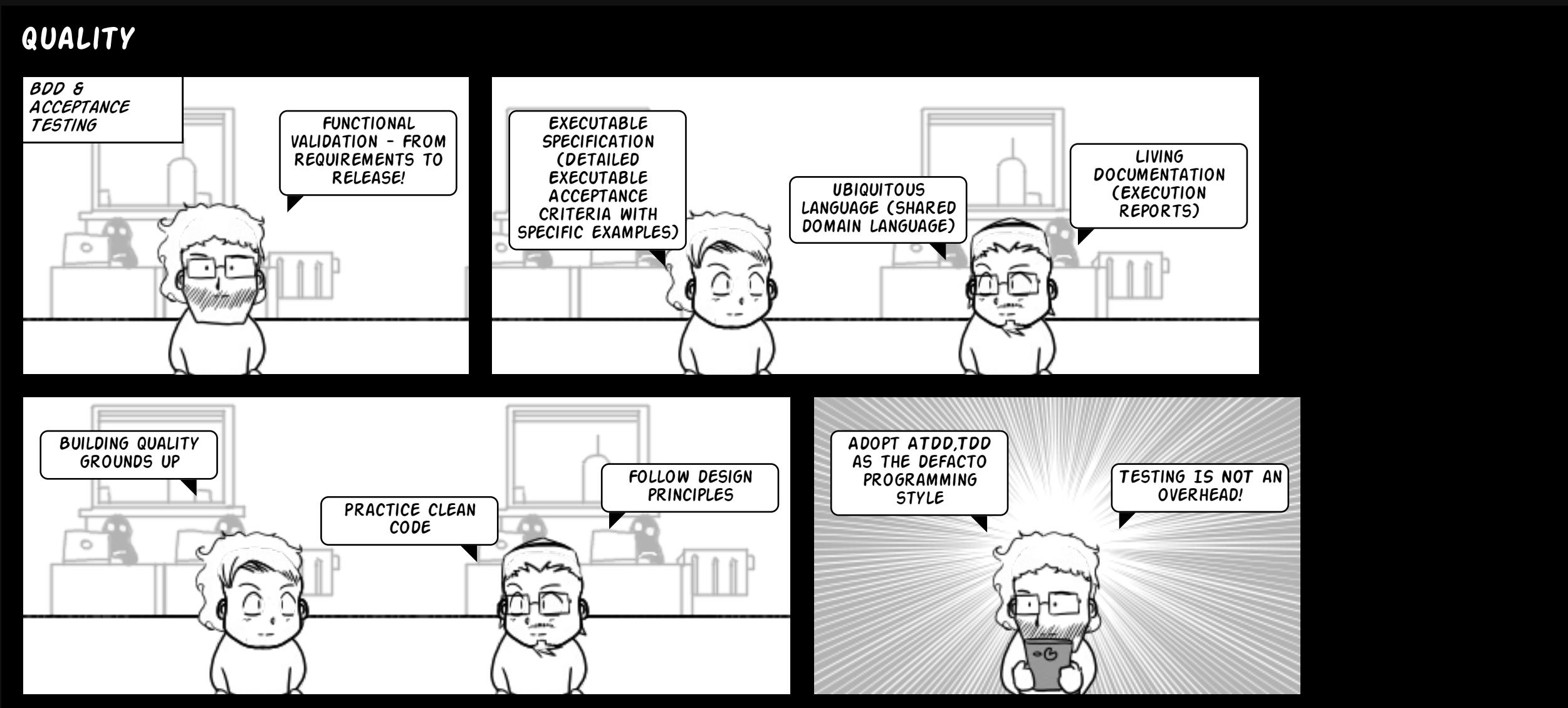
Architecture & Design ...

```
public interface Subject {  
    public void registerObserver(Observer o); }  
    public void removeObserver(Observer o); }  
    public void notifyObservers(); }  
  
public interface Observer {  
    public void update(float temp, float humidity, float pressure); }  
  
public interface DisplayElement {  
    public void display(); }  
  
Both of these methods take an  
Observer as an argument; that is, the  
Observer to be registered or removed.  
  
This method is called to notify all observers  
when the Subject's state has changed.  
  
These are the state values the Observers get from  
the Subject when a weather measurement changes  
  
The Observer interface is  
implemented by all observers,  
so they all have to implement  
the update() method. Here  
we're following Mary and  
Sue's lead and passing the  
measurements to the observers.  
  
The DisplayElement interface just includes one  
method, display(), that we will call when the  
display element needs to be displayed.
```

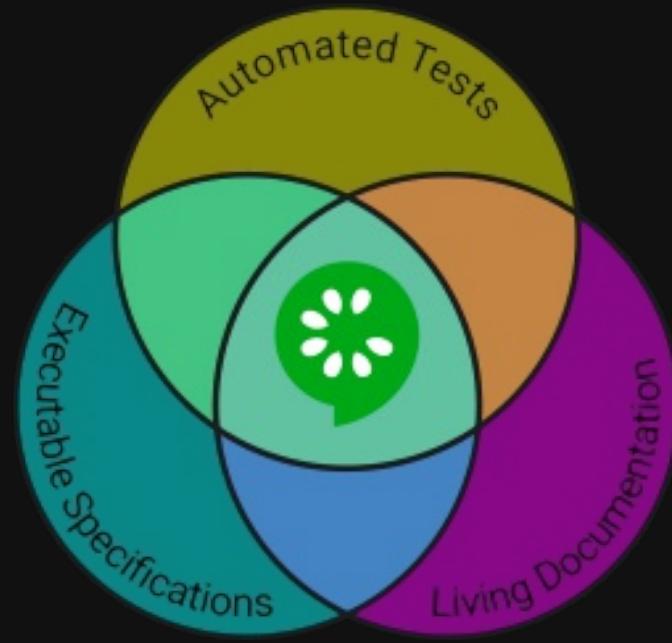
Architecture & Design ...

<https://dzone.com/refcardz/design-patterns>

Ensuring Quality....



Ensuring Quality....



benefits of bdd

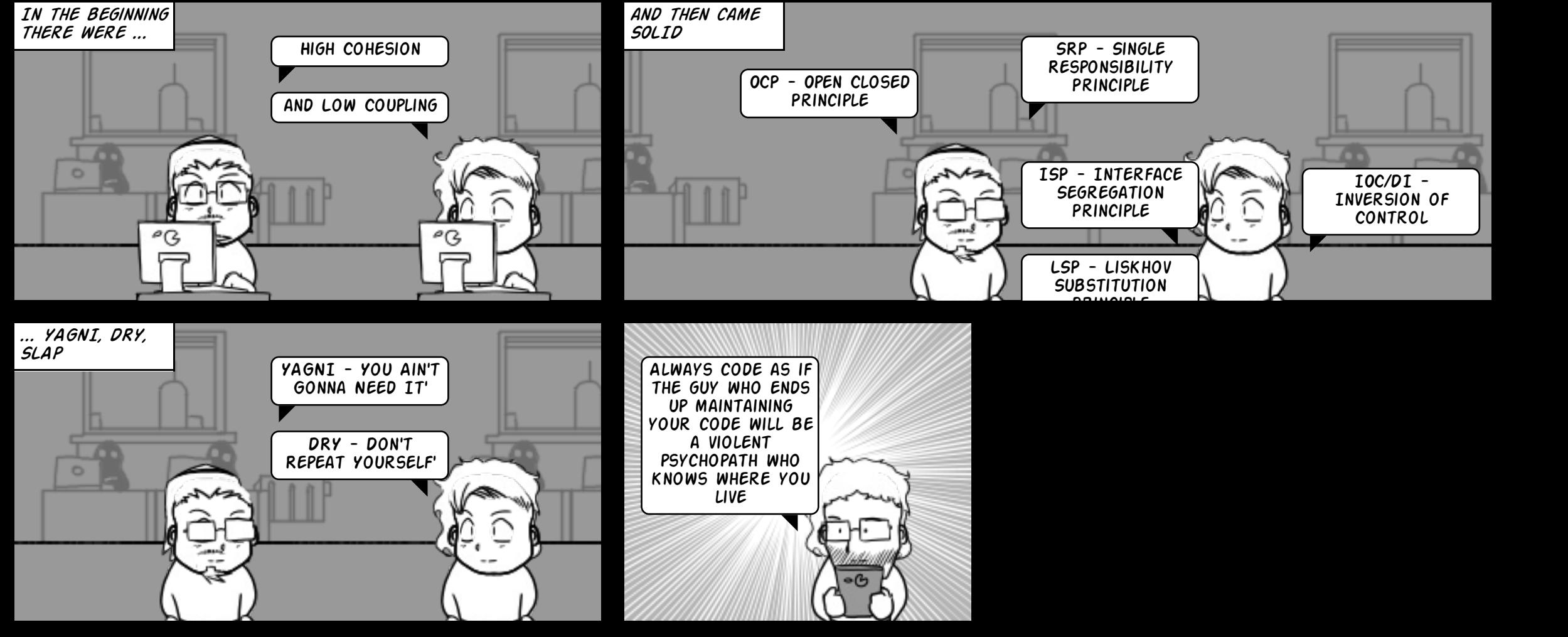
BDD Frameworks

Book: BDD in Action

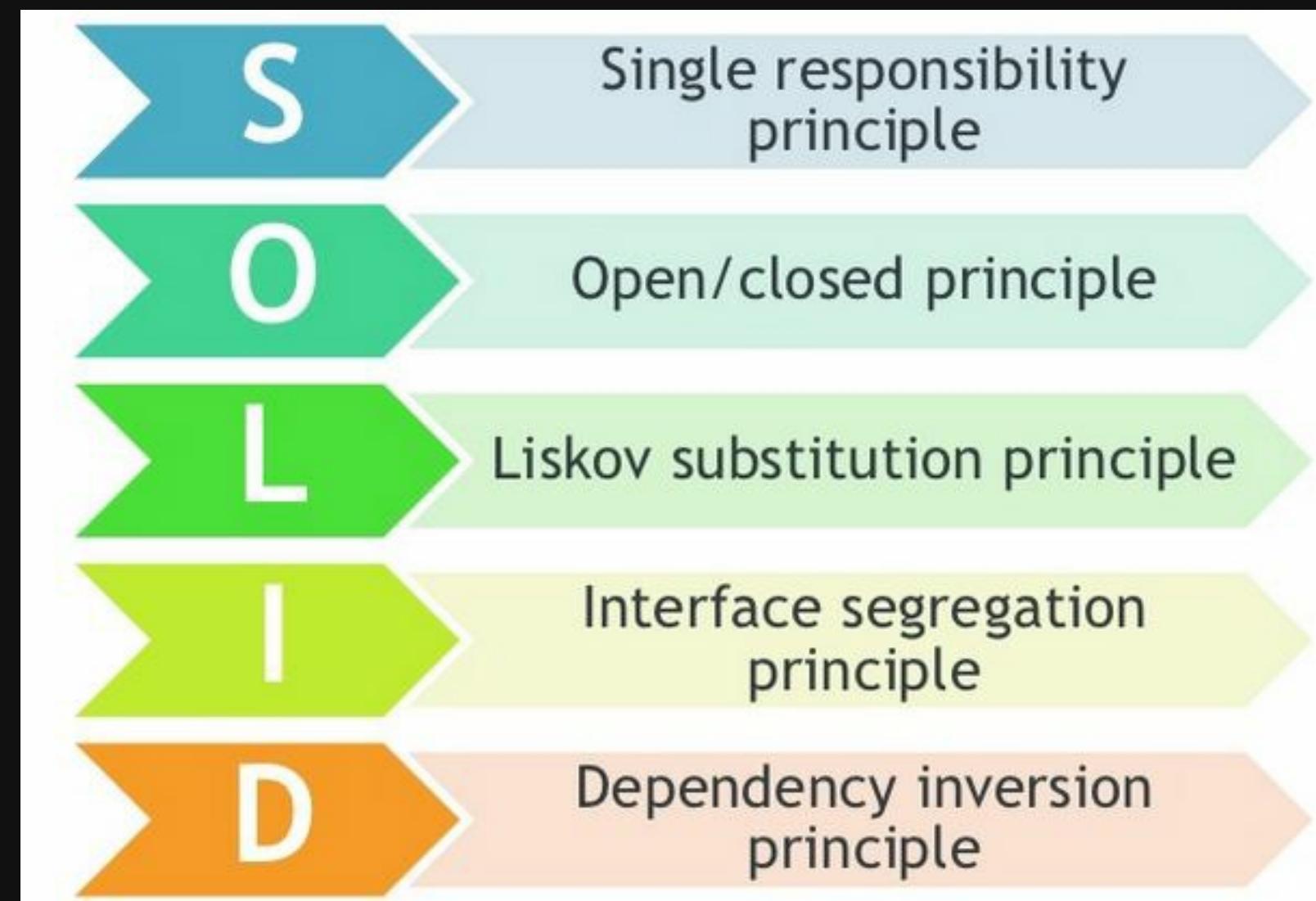
BDD - Introduction from Dan North

Ensuring Quality....

DESIGN PRINCIPLES



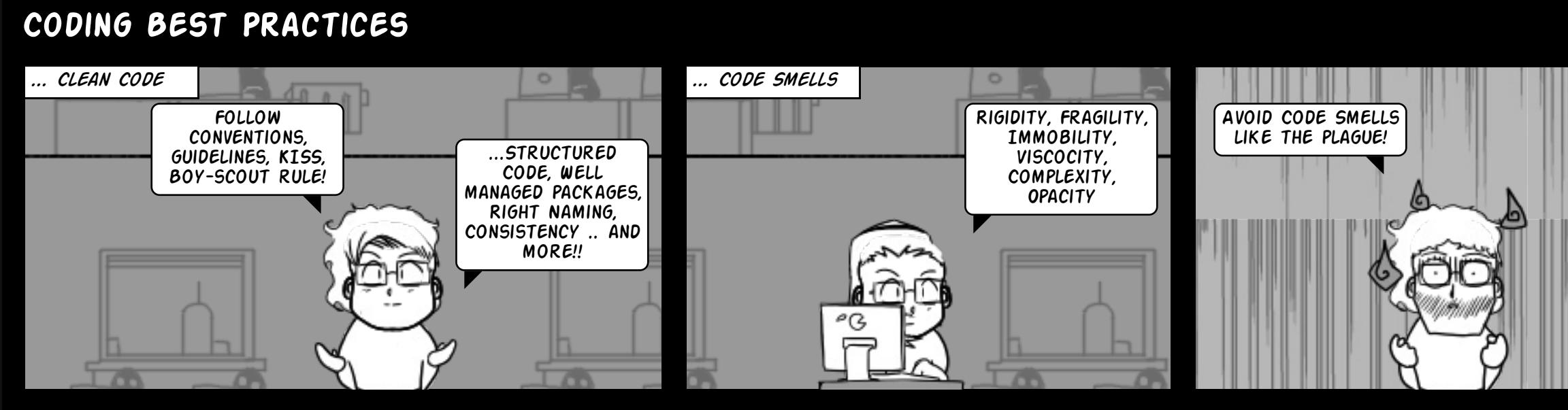
Ensuring Quality....



Ensuring Quality....

<https://leanpub.com/solid/read#leanpub-auto-test-driven-design---tdd>
<http://principles-wiki.net/principles:start>

Ensuring Quality....



Ensuring Quality....

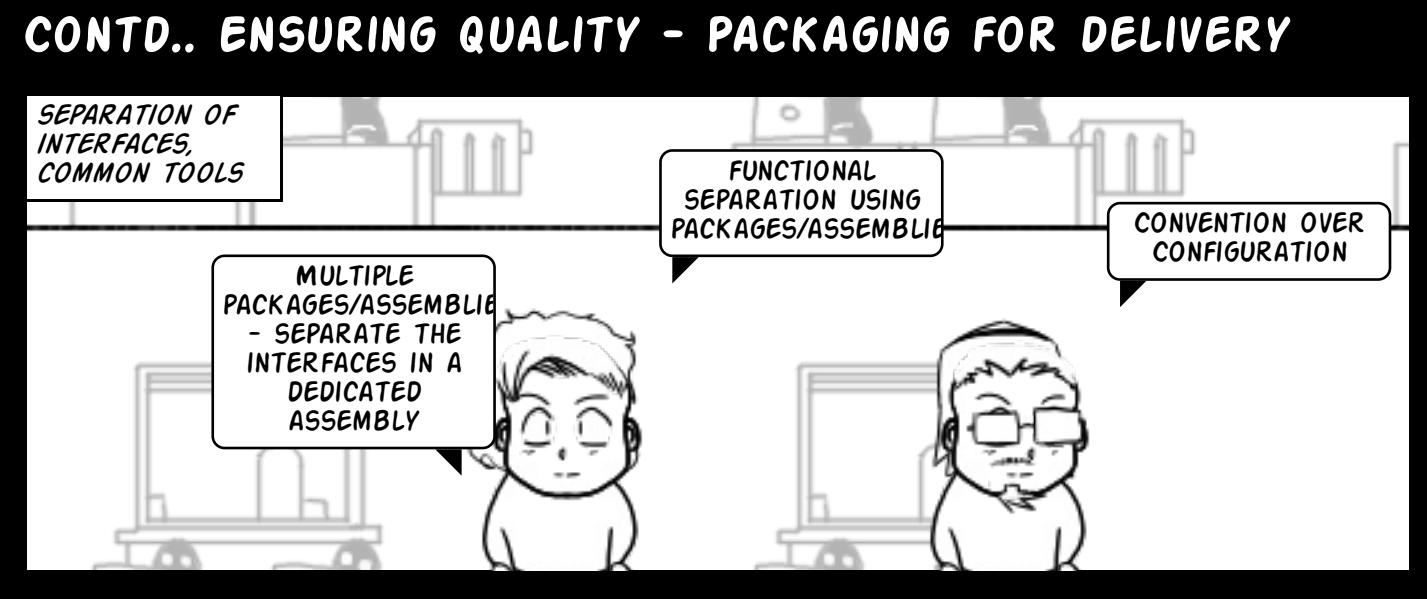
<http://www.planetgeek.ch/wp-content/uploads/2013/06/Clean-Code-V2.1.pdf>

<https://www.amazon.in/Clean-Code-Handbook-Software-Craftsmanship-ebook/dp/B001GSTOAM>

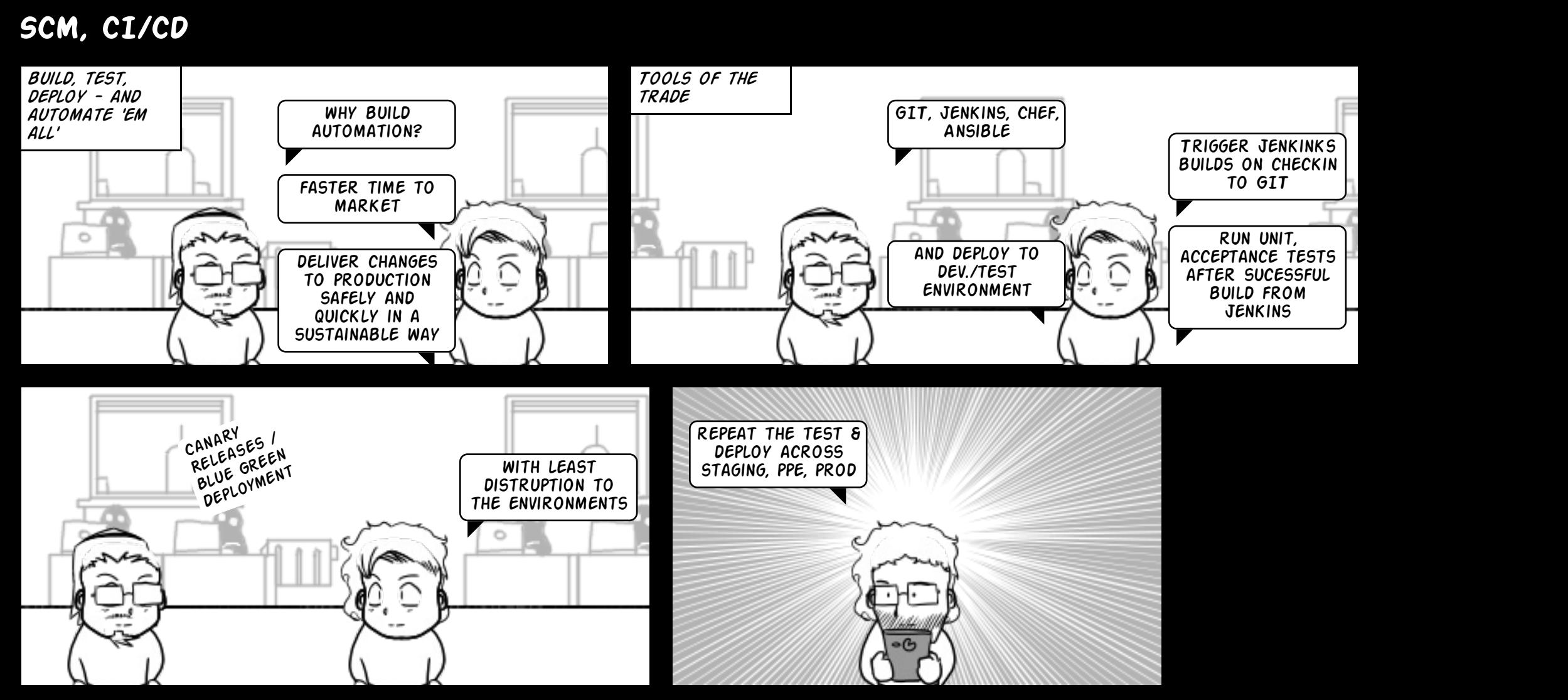
<http://www.hanselman.com/blog/SixEssentialLanguageAgnosticProgrammingBooks.aspx>

<https://github.com/chhantyal/influential-cs-books>

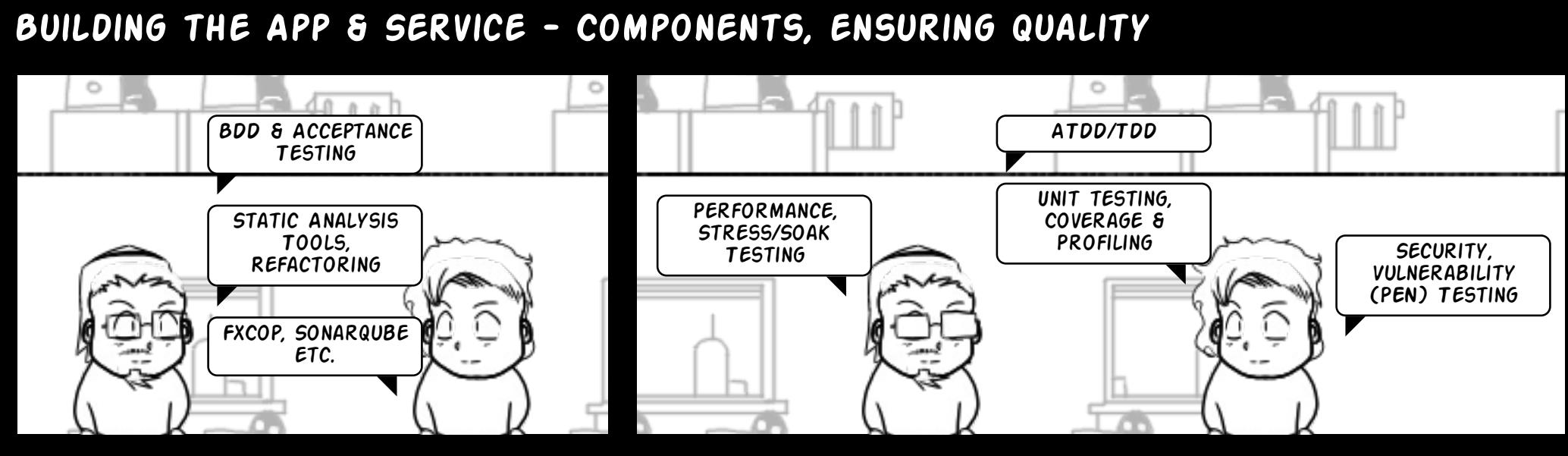
Ensuring Quality....



Ensuring Quality....

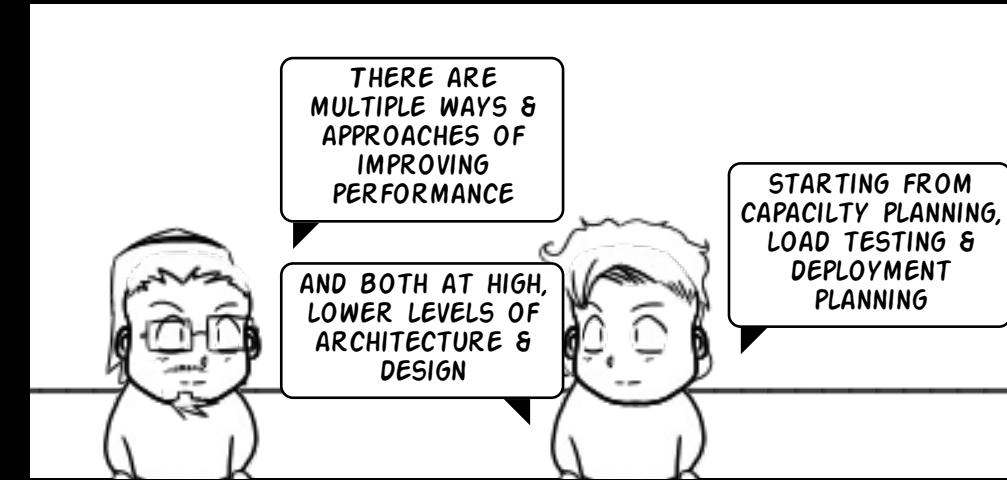
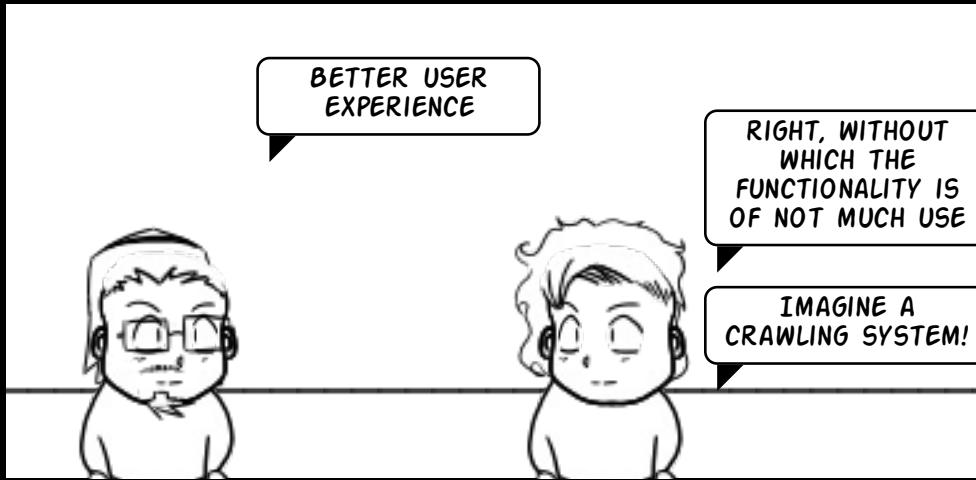


Ensuring Quality....



Performance

WHY IMPROVE PERFORMANCE?



Performance

<http://perftesting.codeplex.com/>

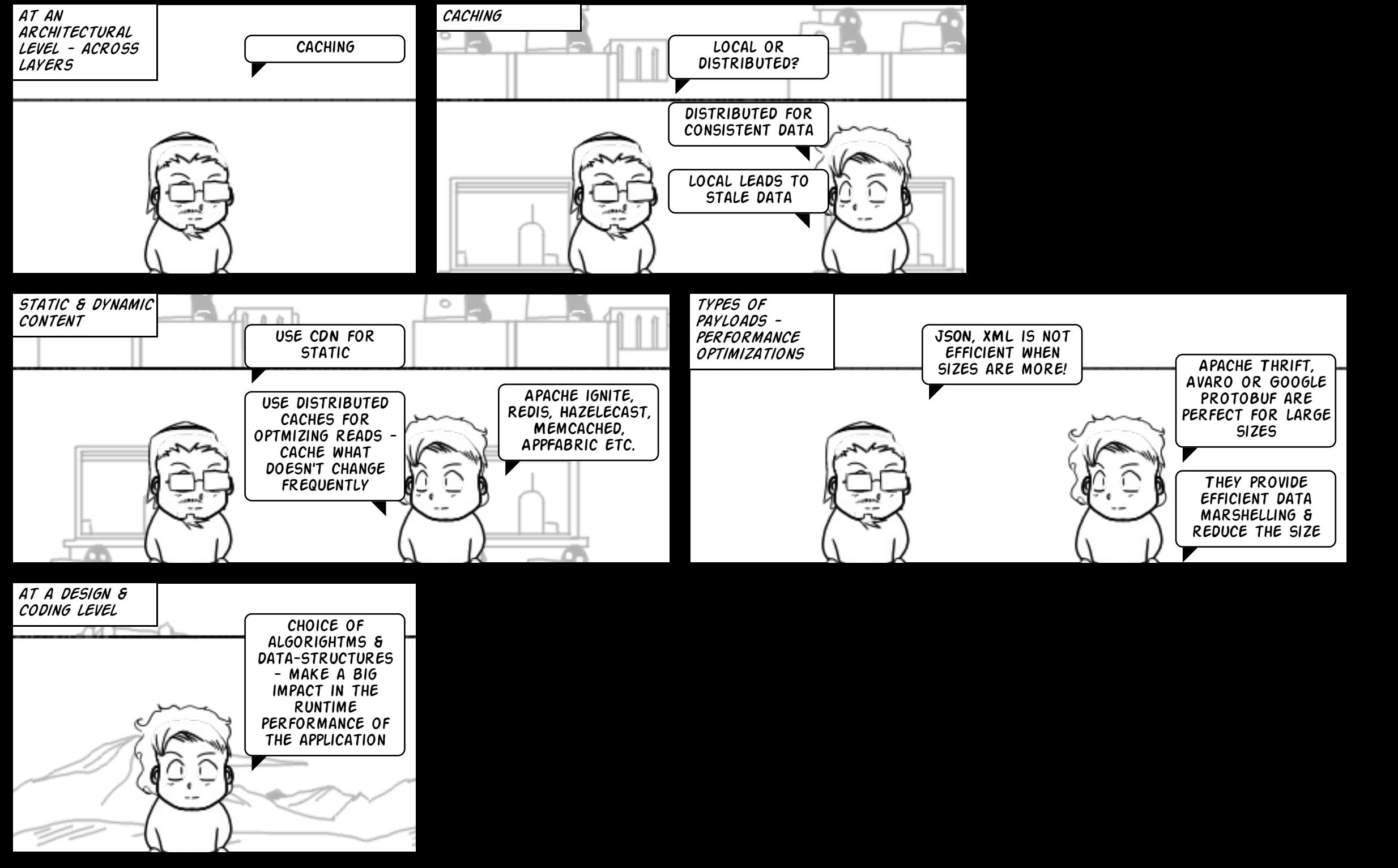
<http://guidanceengineering.codeplex.com/>

<https://msdn.microsoft.com/en-us/library/ms998408.aspx>

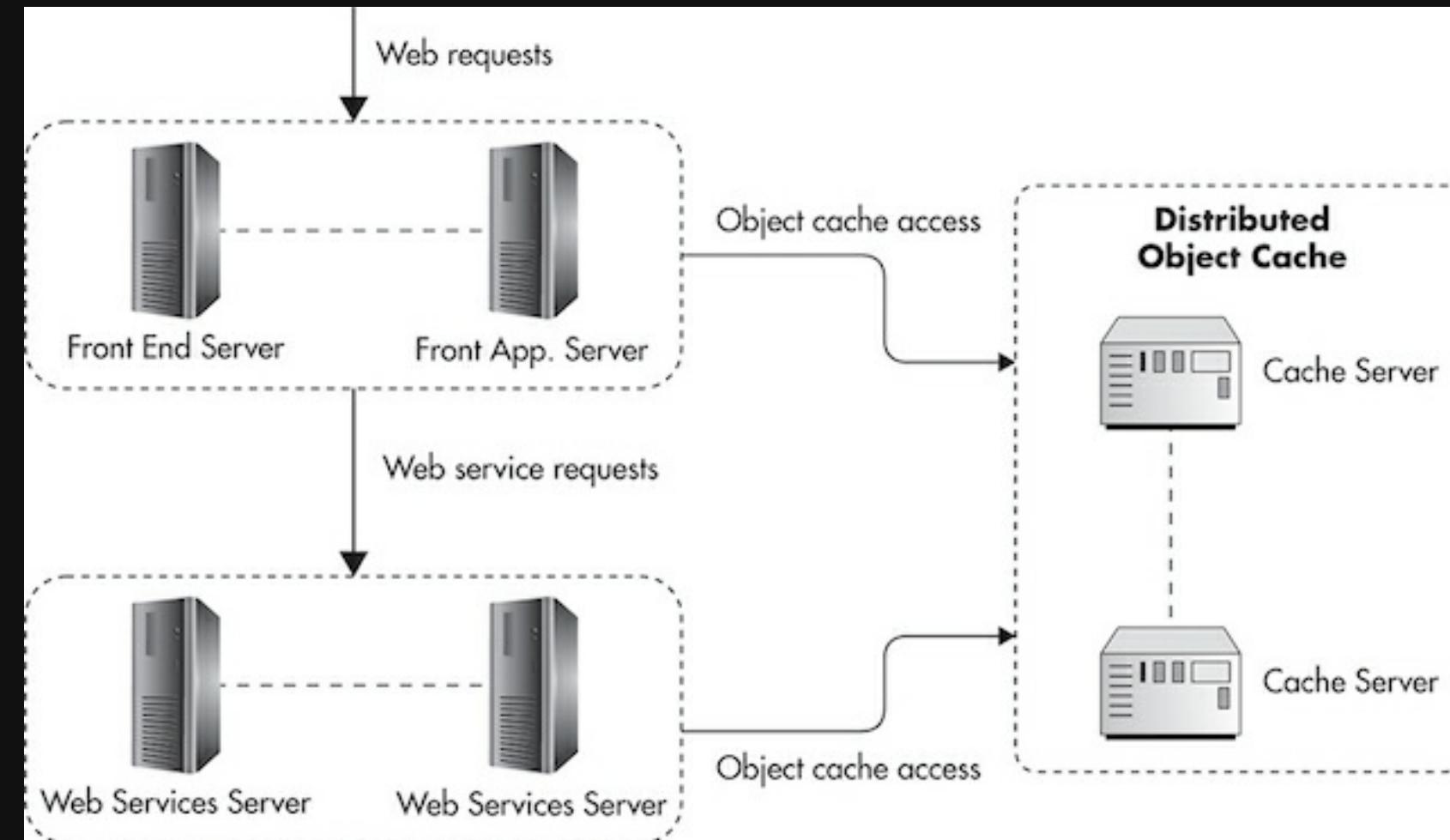
<https://msdn.microsoft.com/practices>

Performance

HOW TO IMPROVE PERFORMANCE

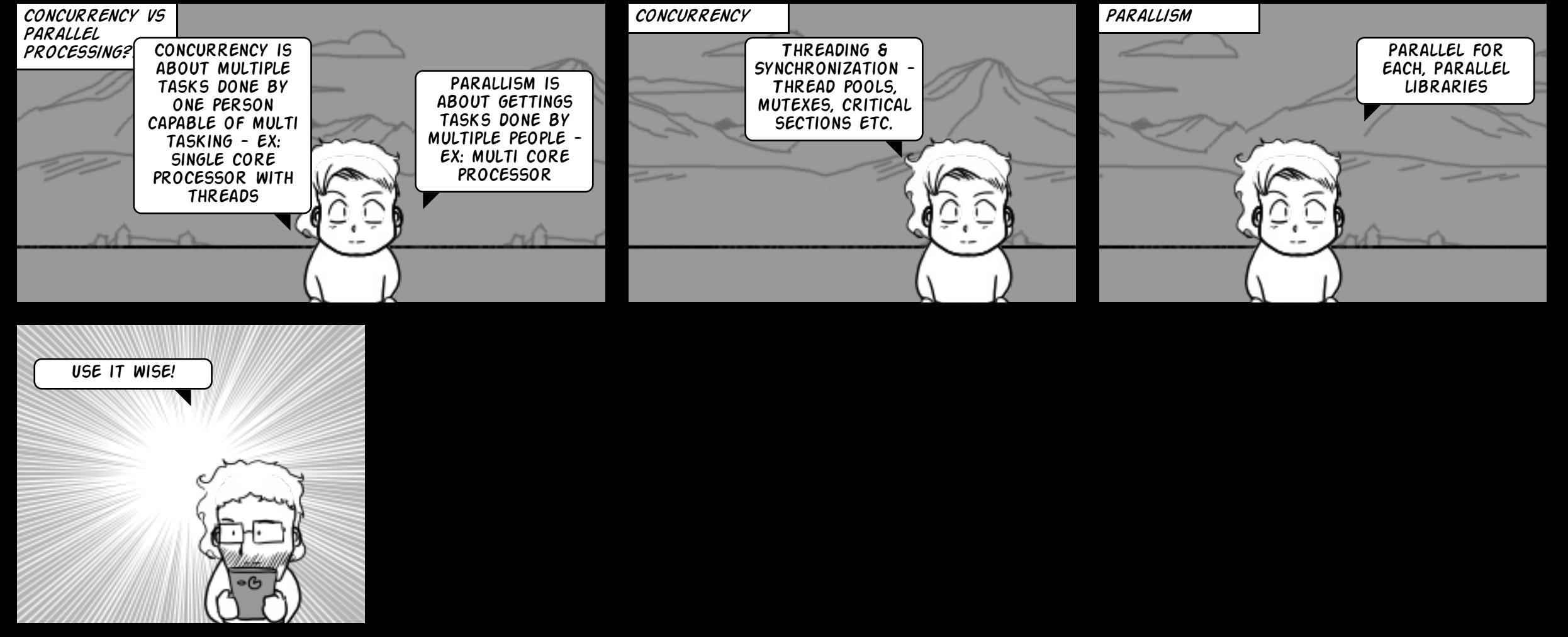


Performance

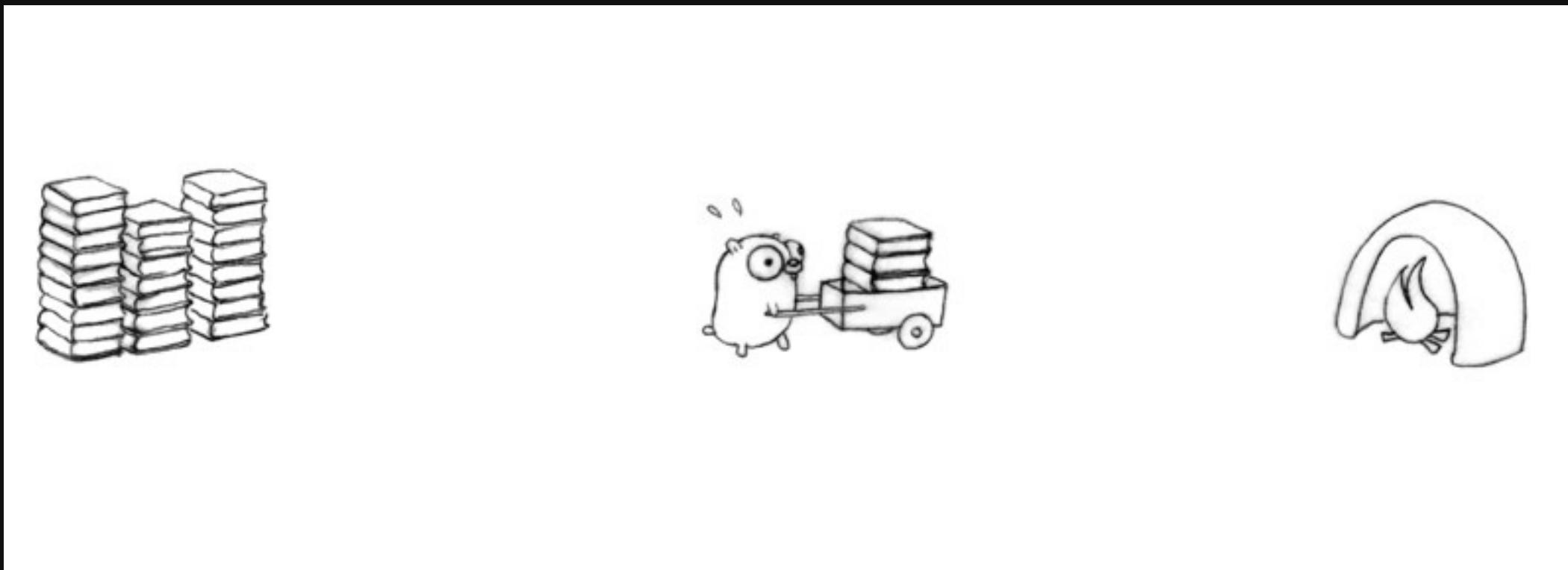


Performance

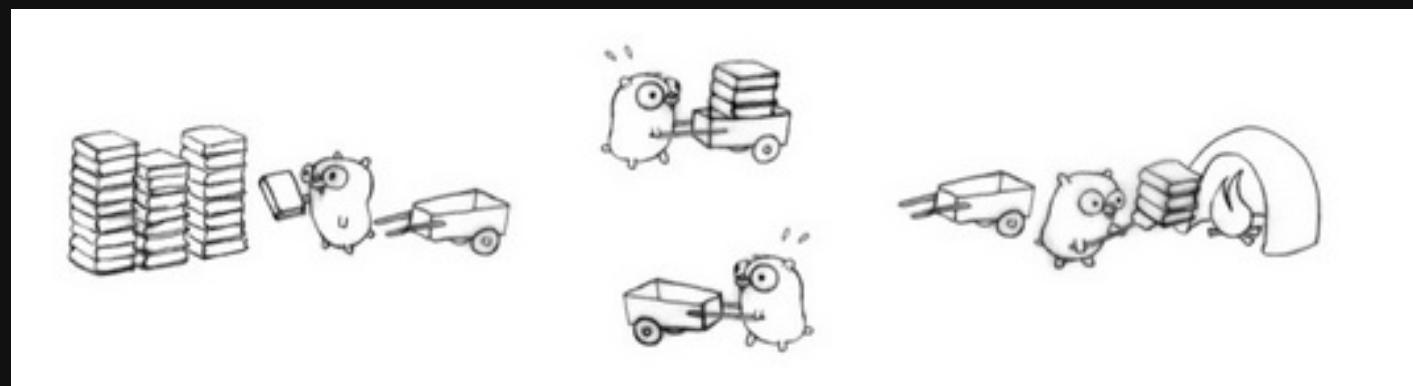
CONCURRENCY & PARALLELISM



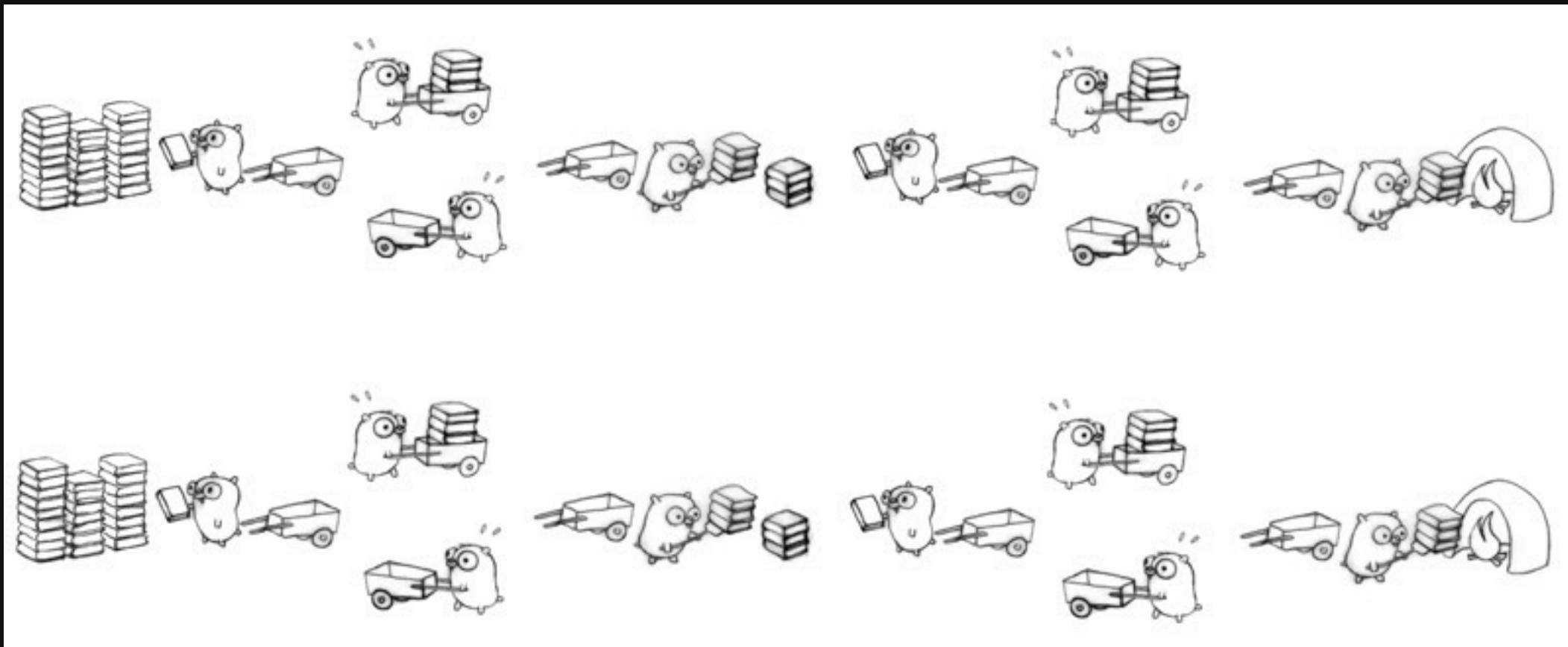
Performance



Performance

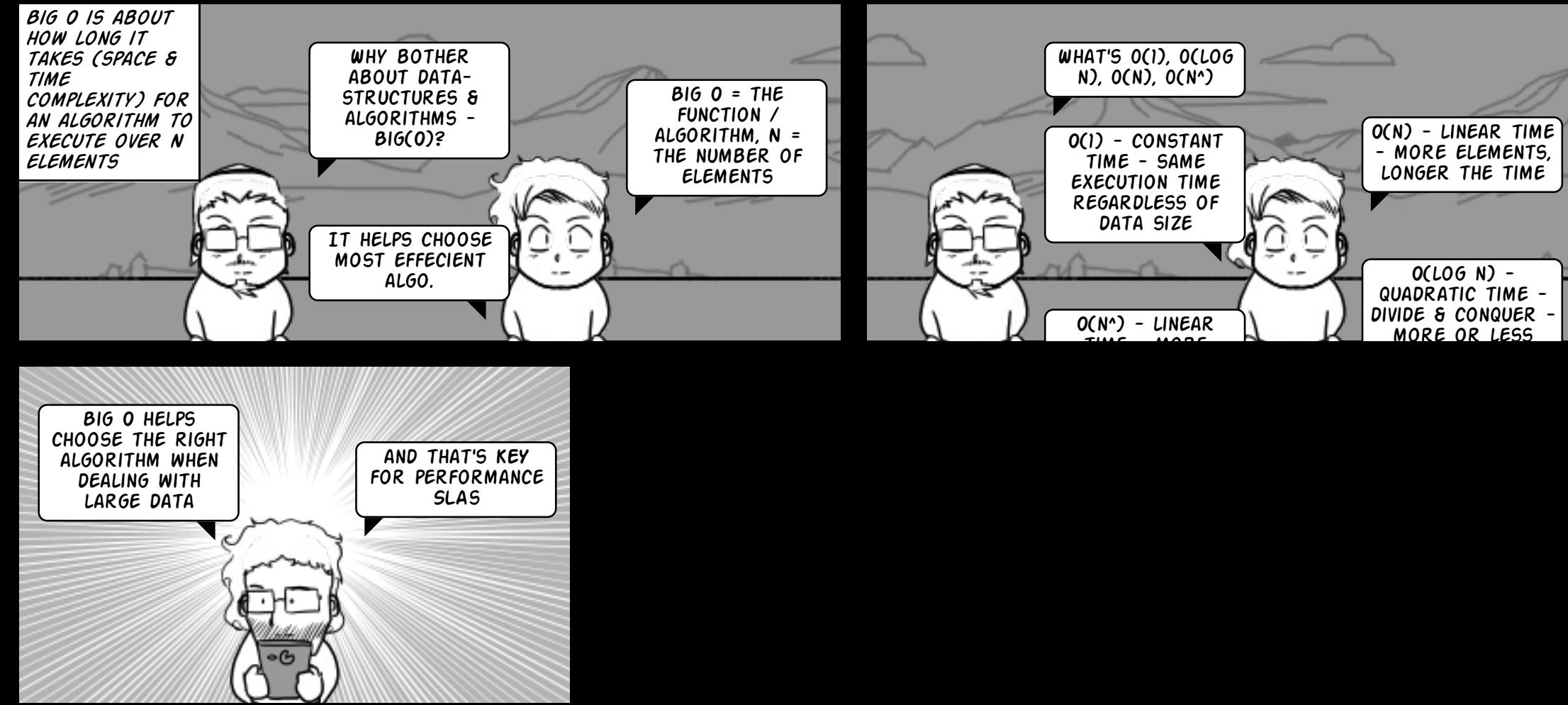


Performance



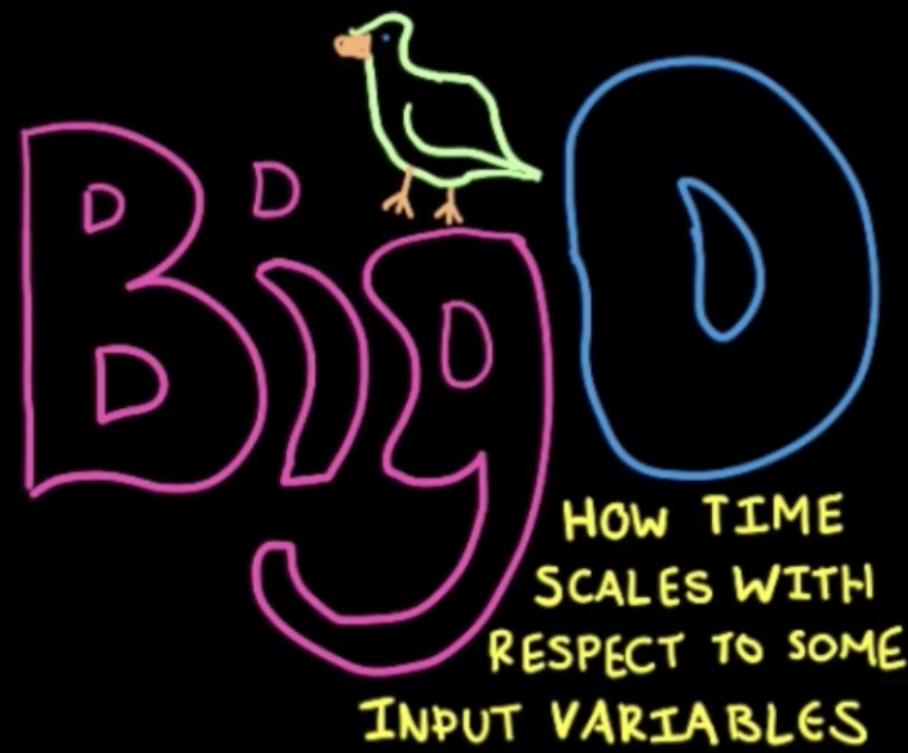
Performance

OPTIMIZATIONS - SPACE & TIME COMPLEXITY



Performance

④ Drop non-dominant terms



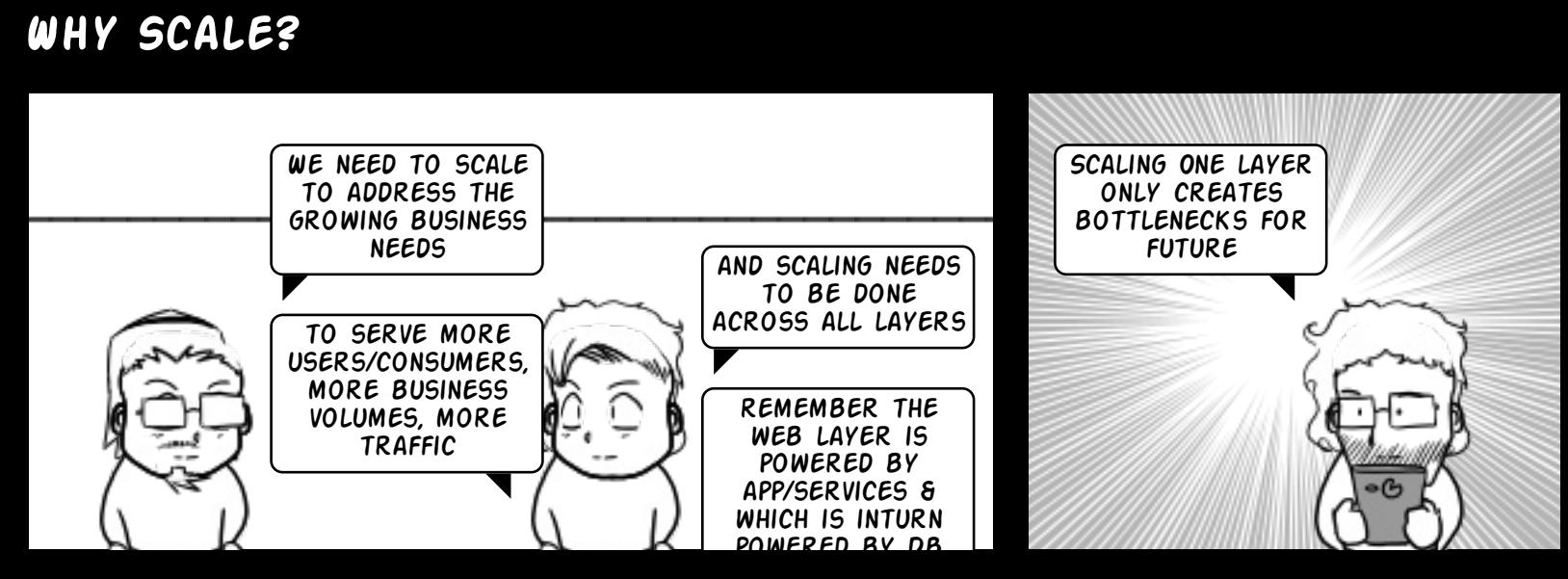
- ① Different steps get added
- ② Drop constants
- ③ Different inputs \Rightarrow different variables

```
function whyWouldIDoThis(array){  
    max = NULL  
    O(n) {  
        for each a in array {  
            max = MAX(a, max)  
        }  
        print max  
    }  
    O(n2) {  
        for each a in array {  
            for each b in array {  
                print a,b  
            }  
        }  
    }  
    O(n2) ≤ O(n+n2) ≤ O(n2+n2)  
    *if LEFT and RIGHT are equivalent  
    (see RULE 2), then CENTER is too*  
    O(n+n2)  $\Rightarrow$  O(n2)
```

Performance

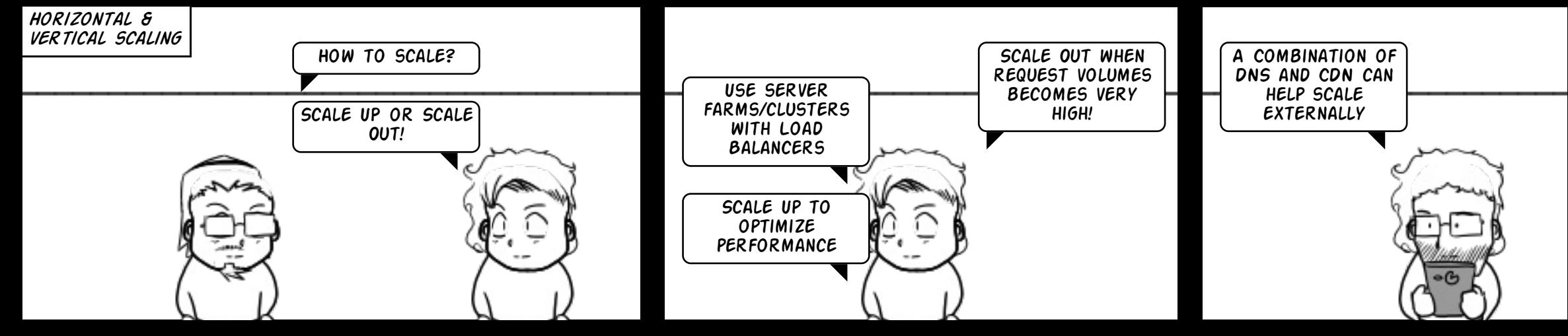
<https://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/>
https://www.youtube.com/watch?v=_vX2sjlpXU
<https://www.youtube.com/watch?v=v4cd1O4zkGw>
https://www.youtube.com/watch?v=-Eiw_-v__Vo
<http://bigocheatsheet.com/>
https://en.wikipedia.org/wiki/Big_O_notation
<http://adrianmejia.com/blog/2014/02/13/algorithms-for-dummies-part-1-sorting/>
<http://algorithmiccomplexity.com/>
http://web.mit.edu/16.070/www/lecture/big_o.pdf

Scalability

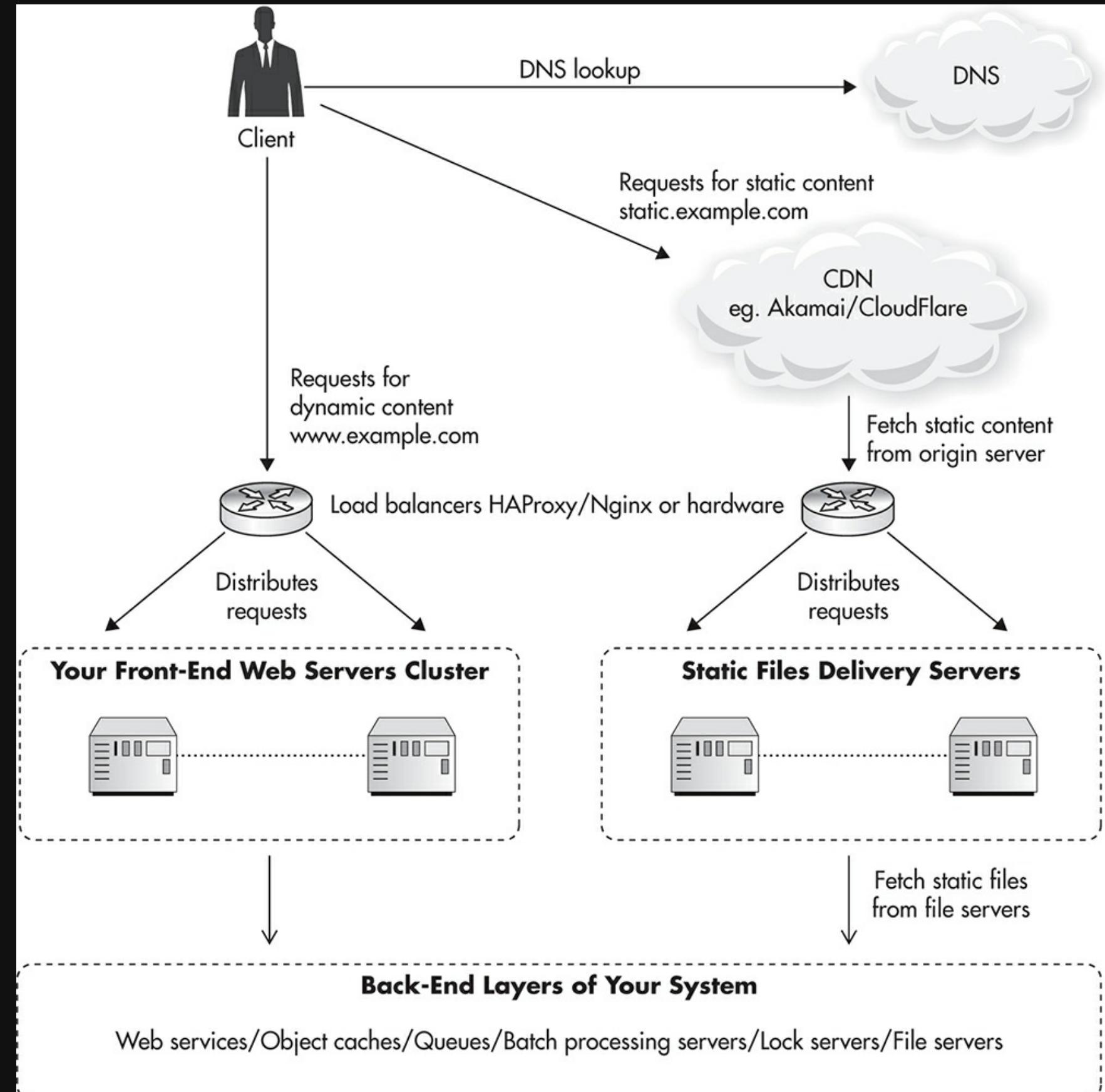


Scalability

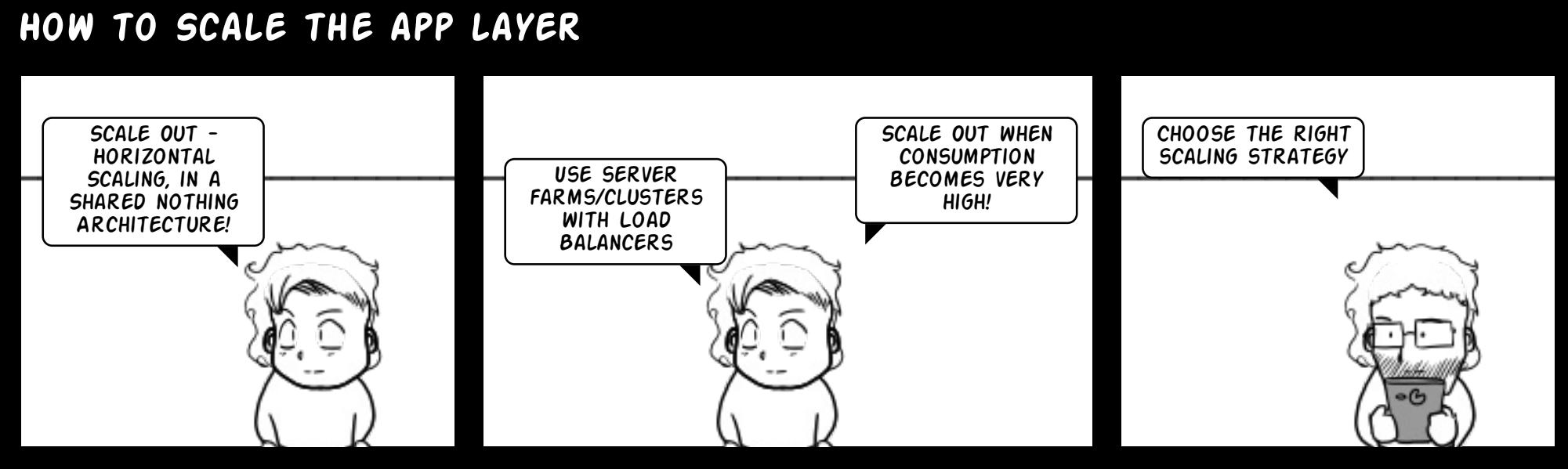
HOW TO SCALE THE WEB LAYER



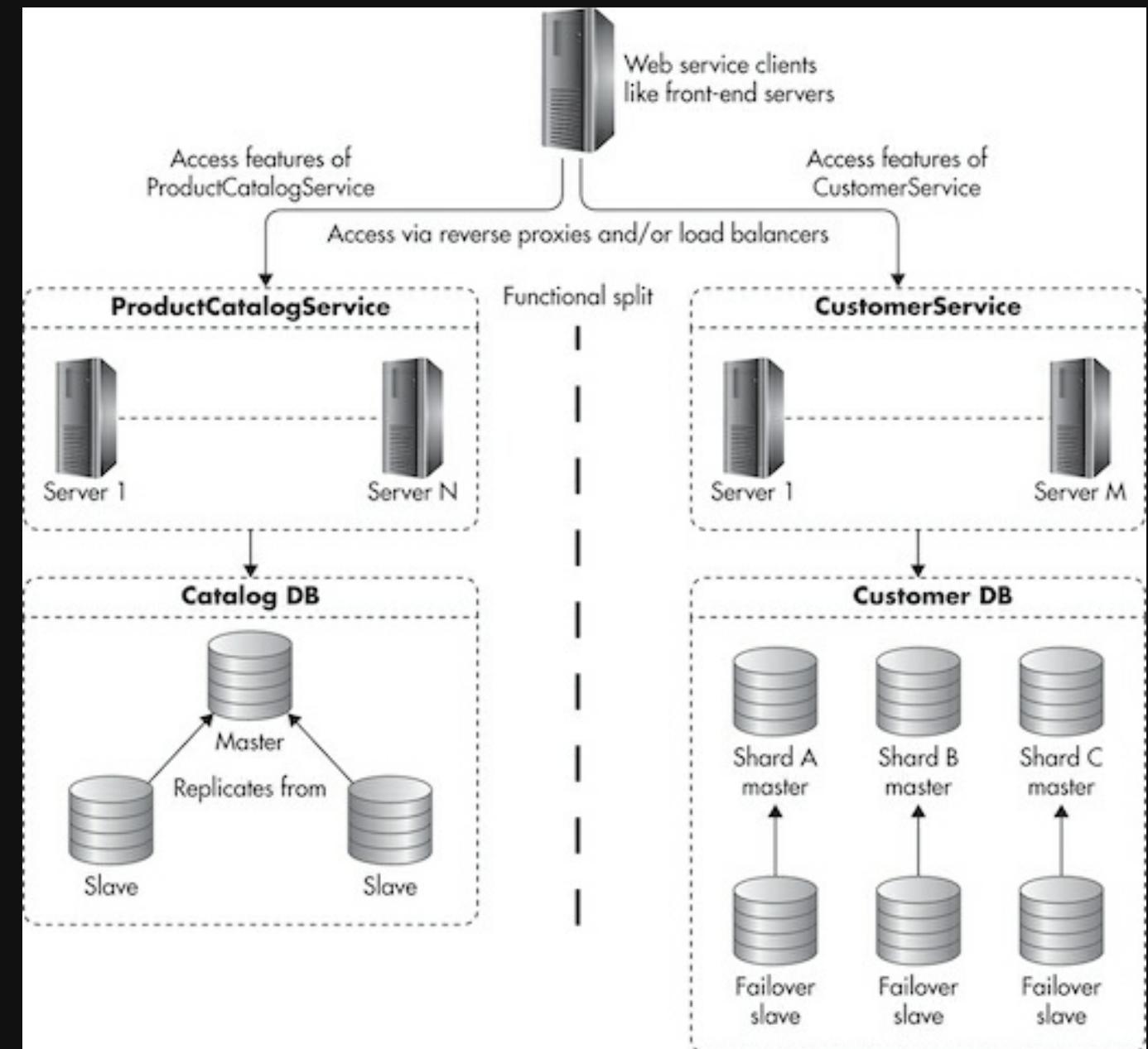
Scalability



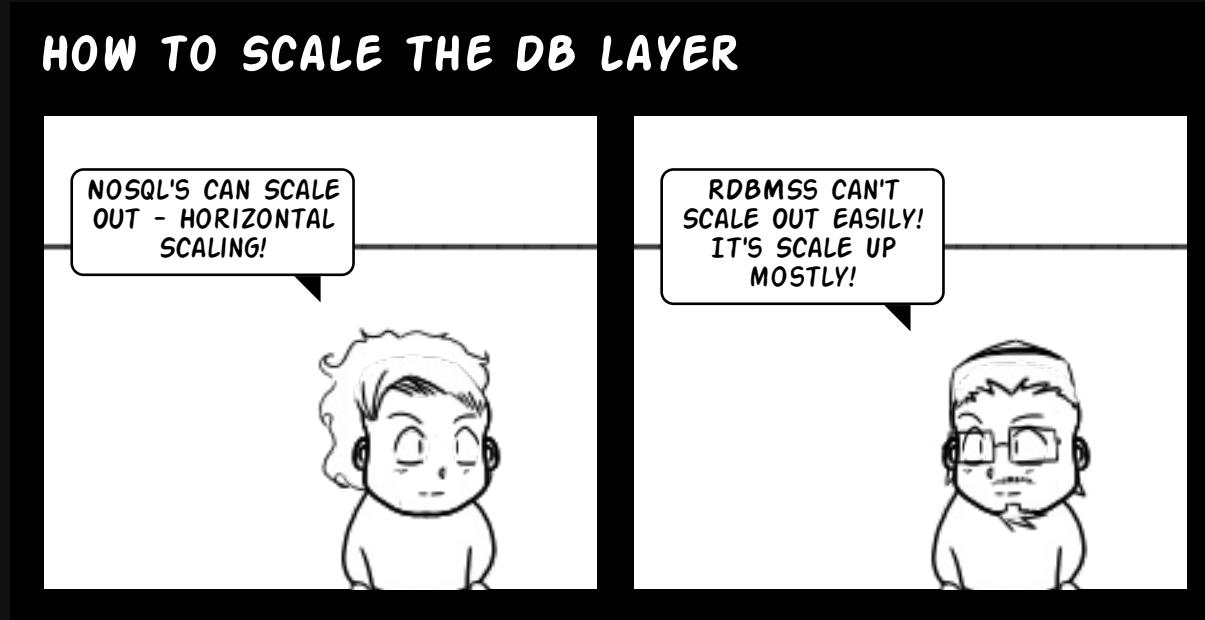
Scalability



Scalability

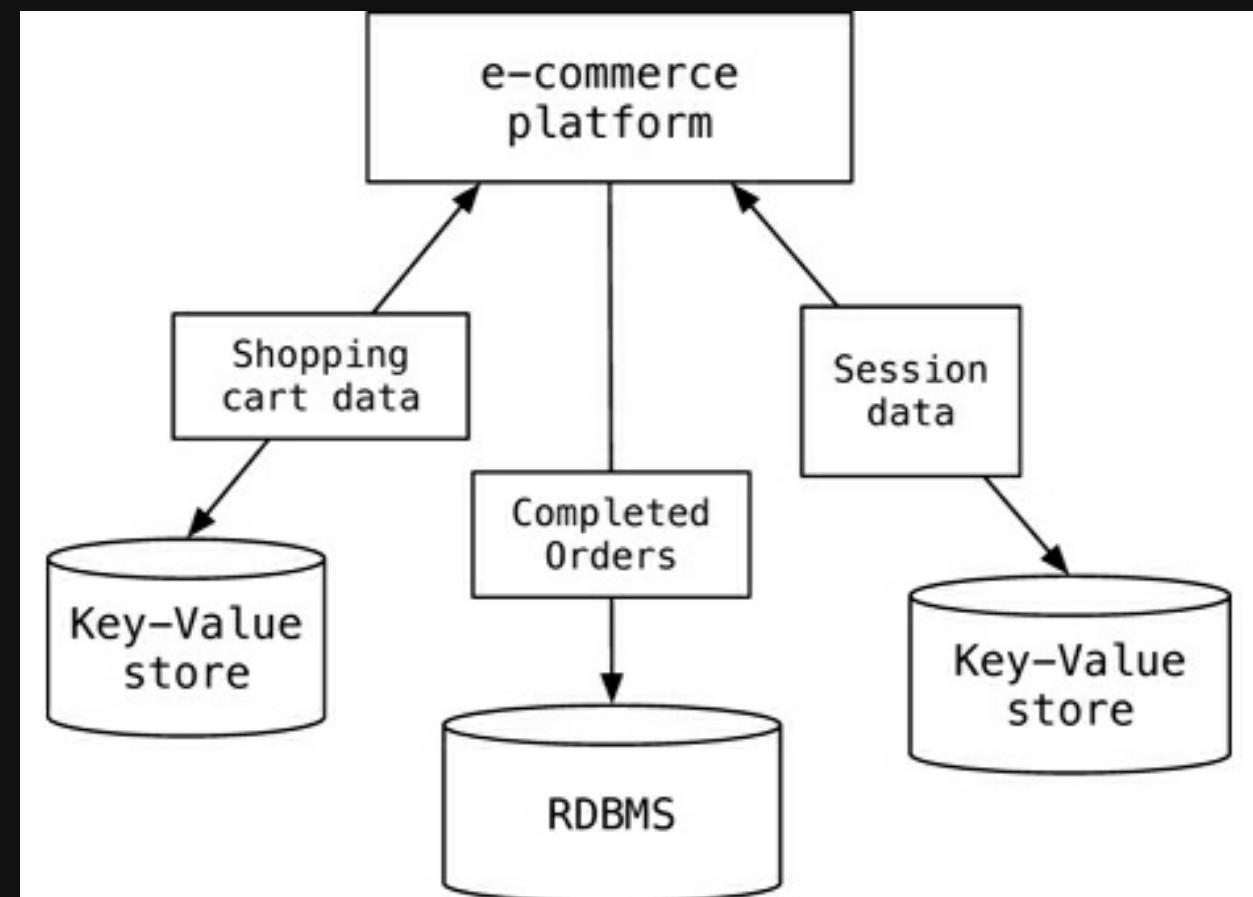


Scalability



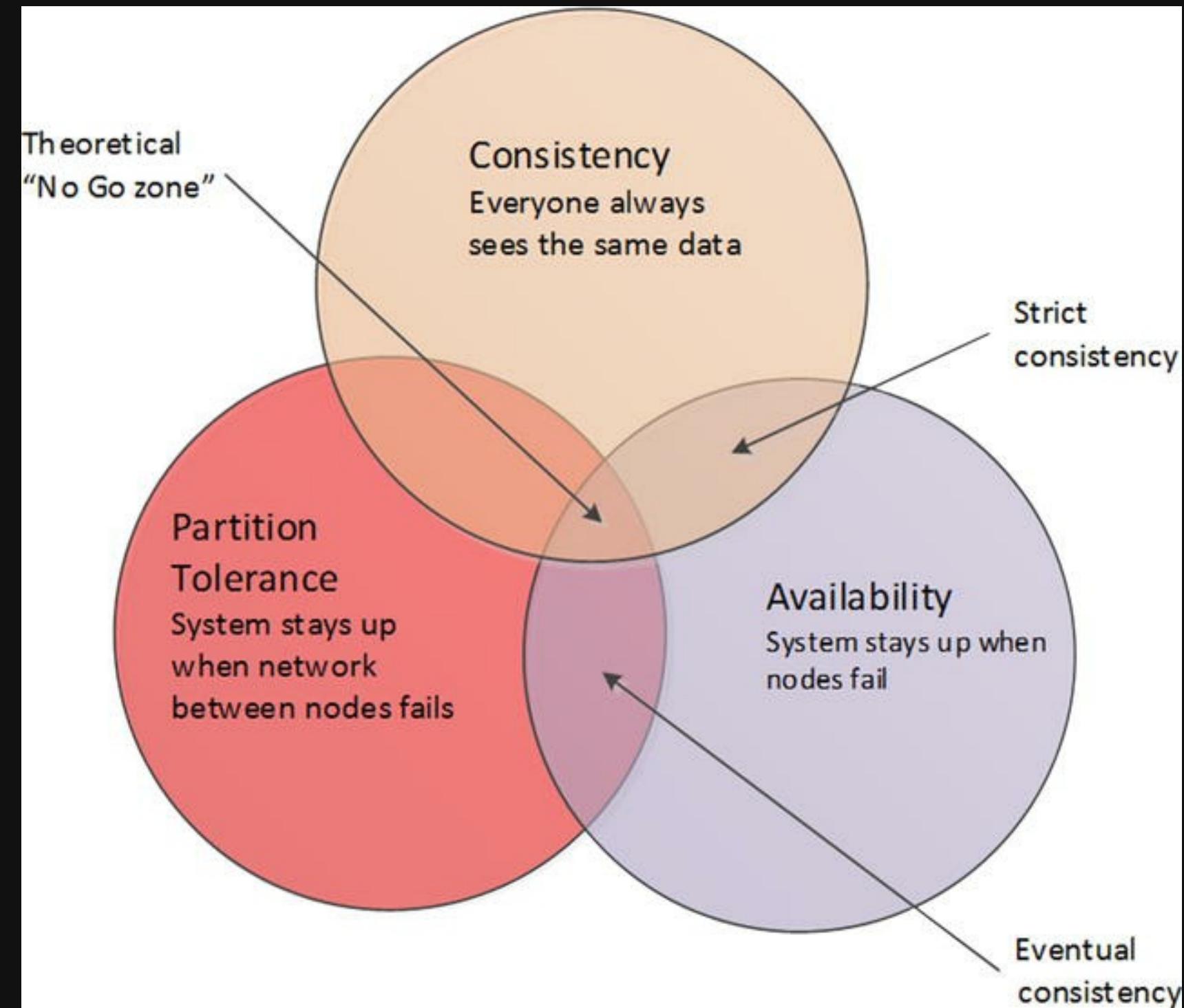
Scalability

Scale Database Layer



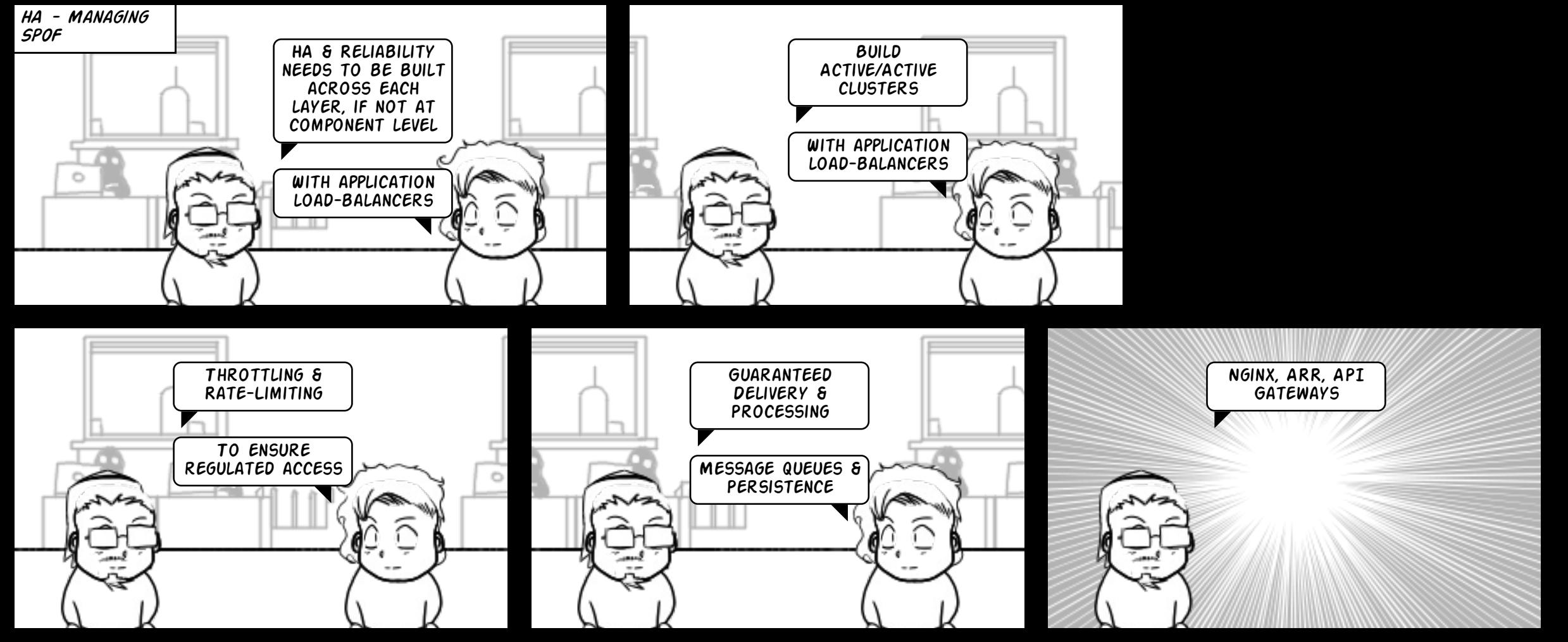
Scalability

CAP Theorem Revisited



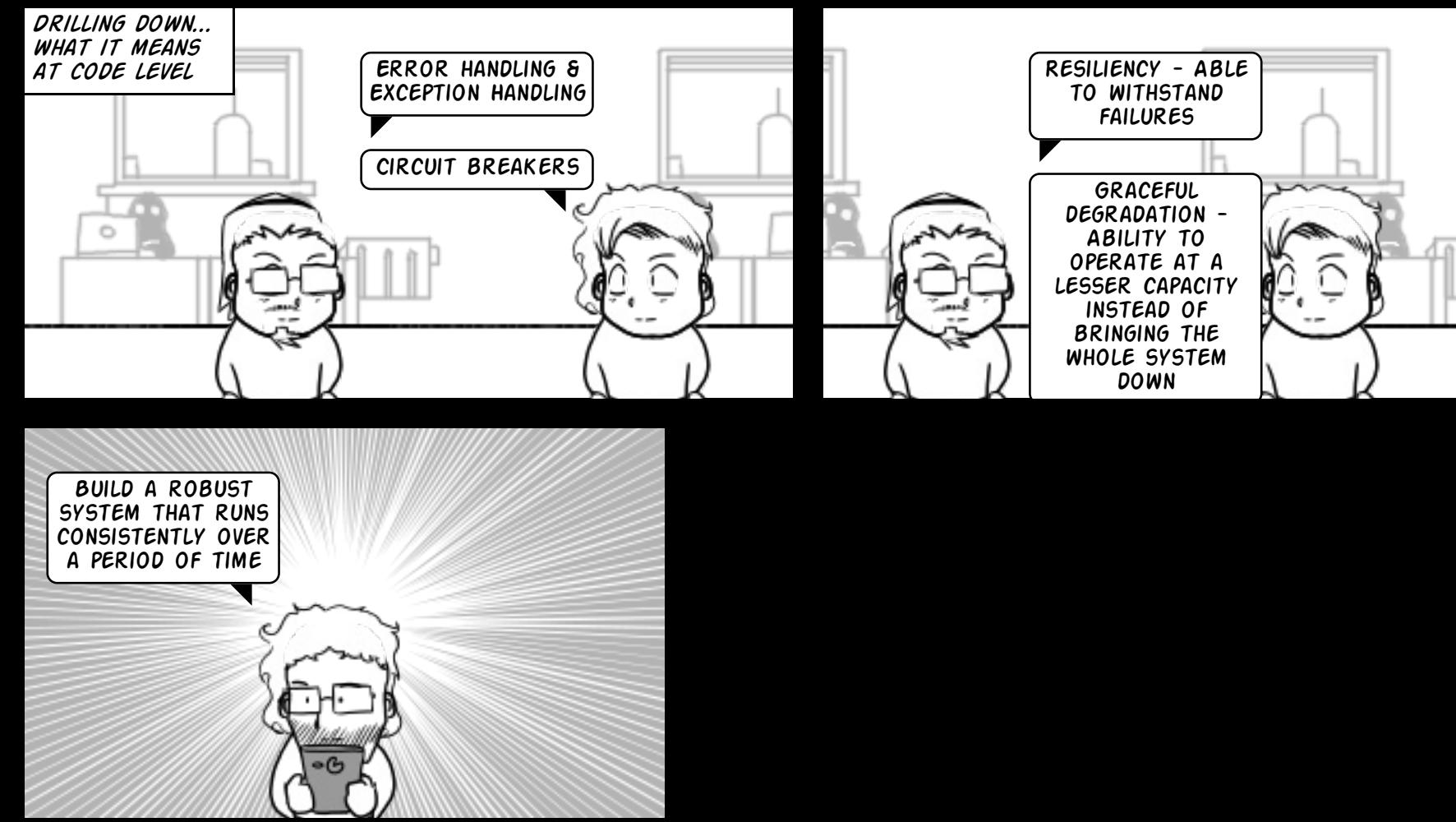
HA & Reliability

HOW TO ENSURE HIGH AVAILABILITY OF THE WEB LAYER



HA & Reliability

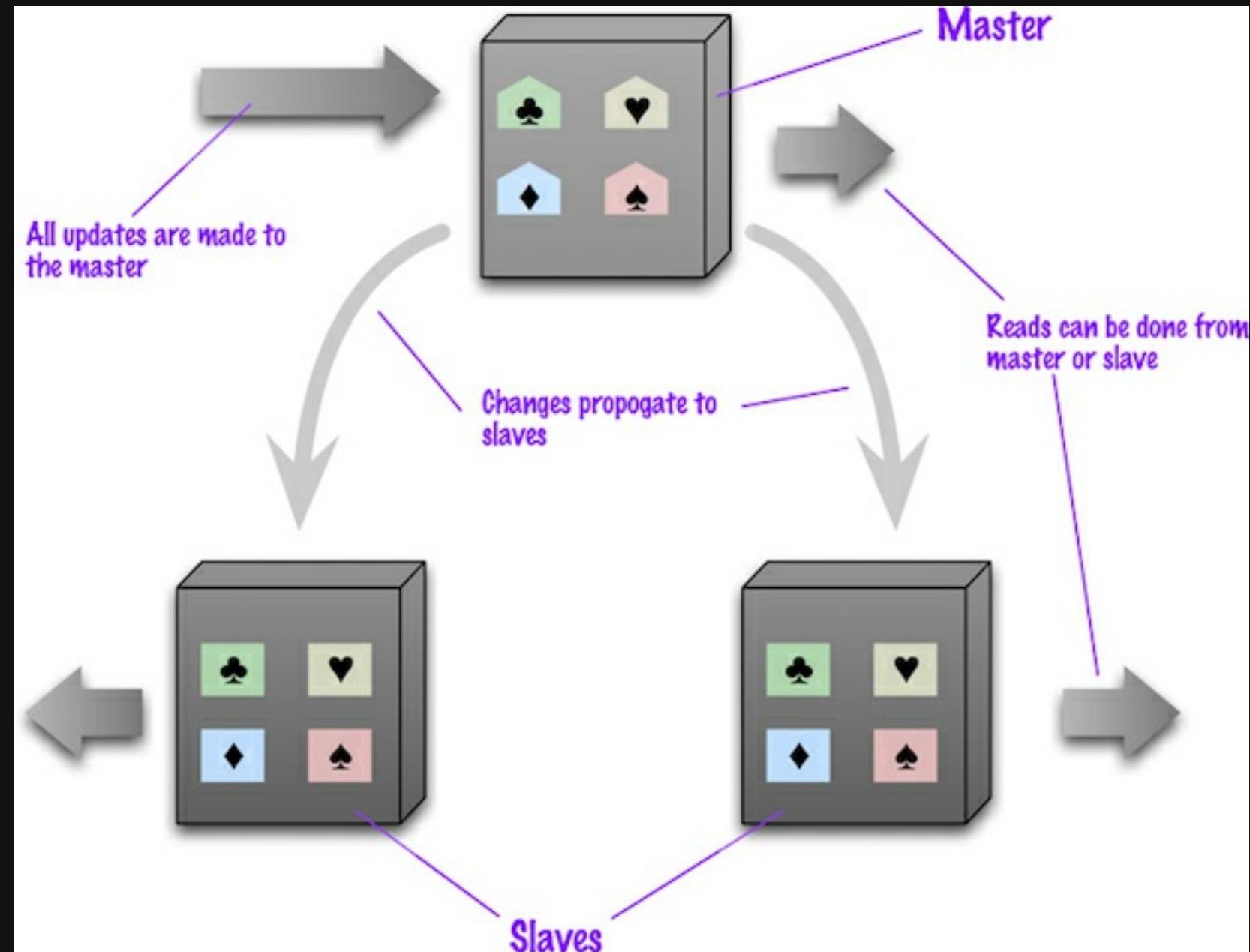
RELIABILITY, RESILIENCY & ROBUSTNESS



HA & Reliability

Master Slave

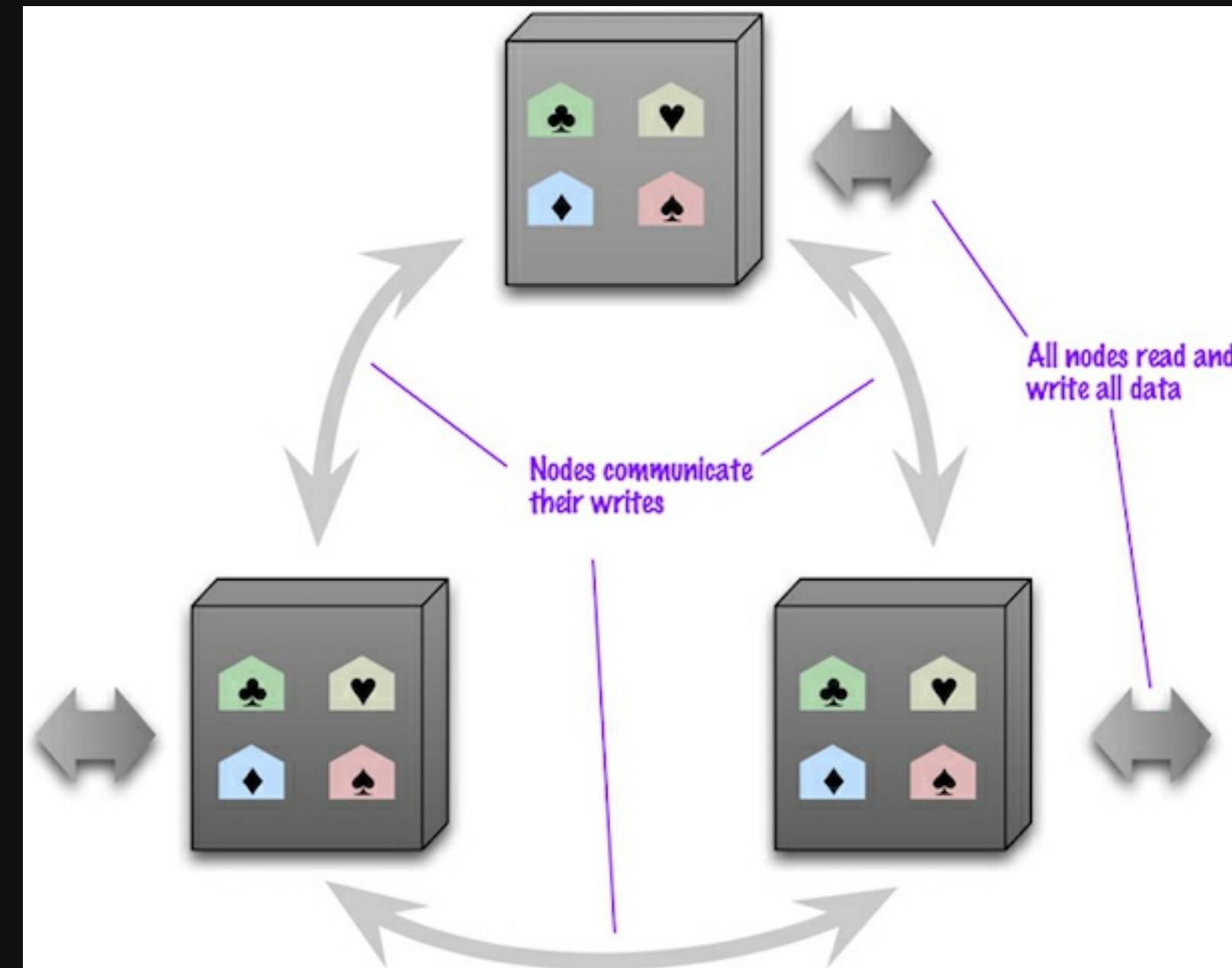
Master Slave replication



HA & Reliability

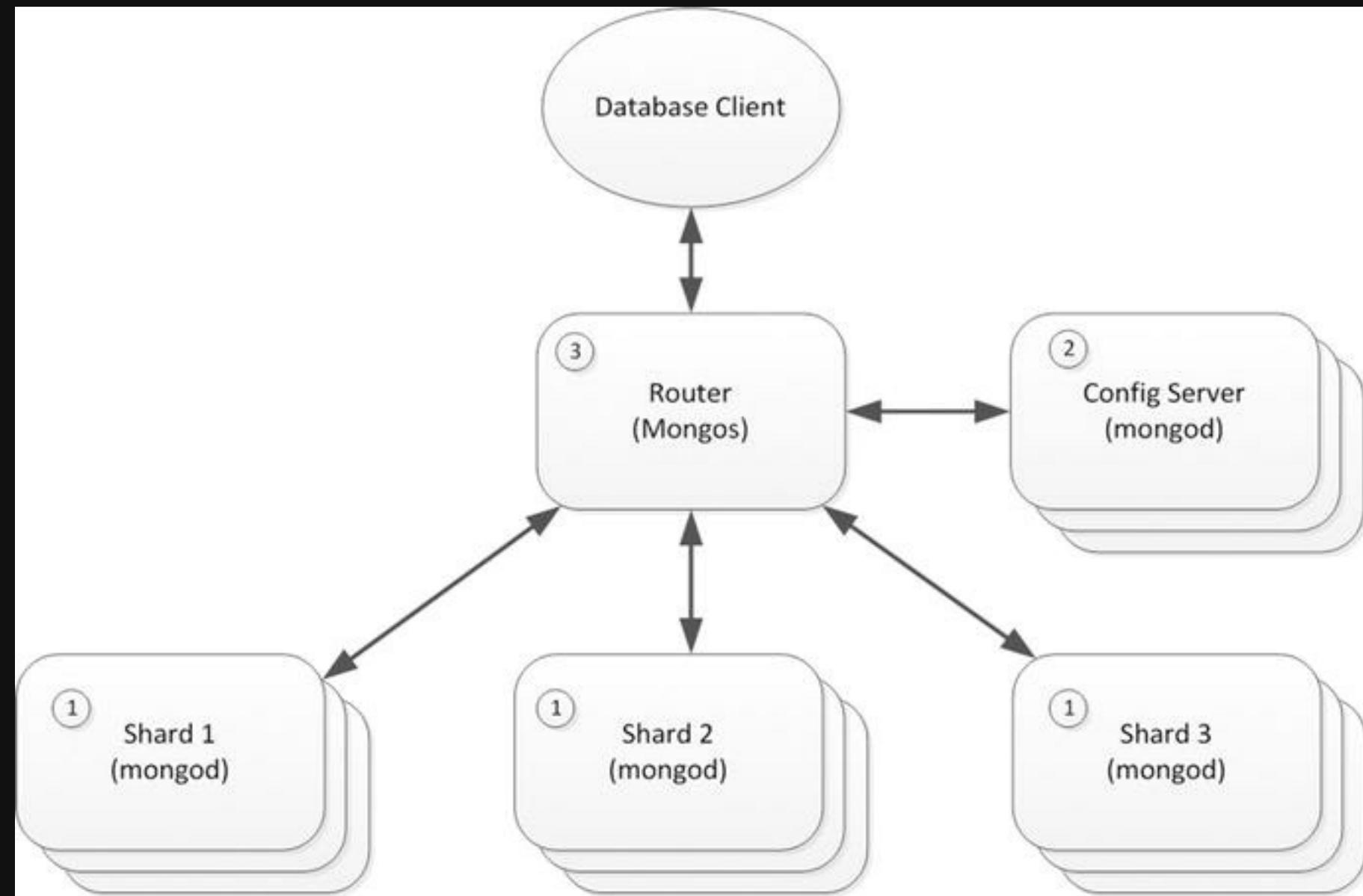
Master Slave

Peer to Peer replication



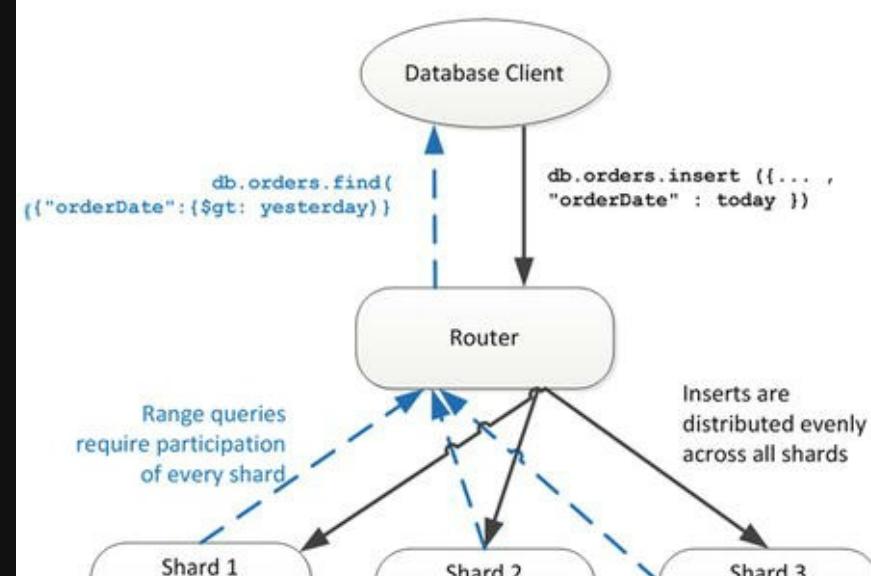
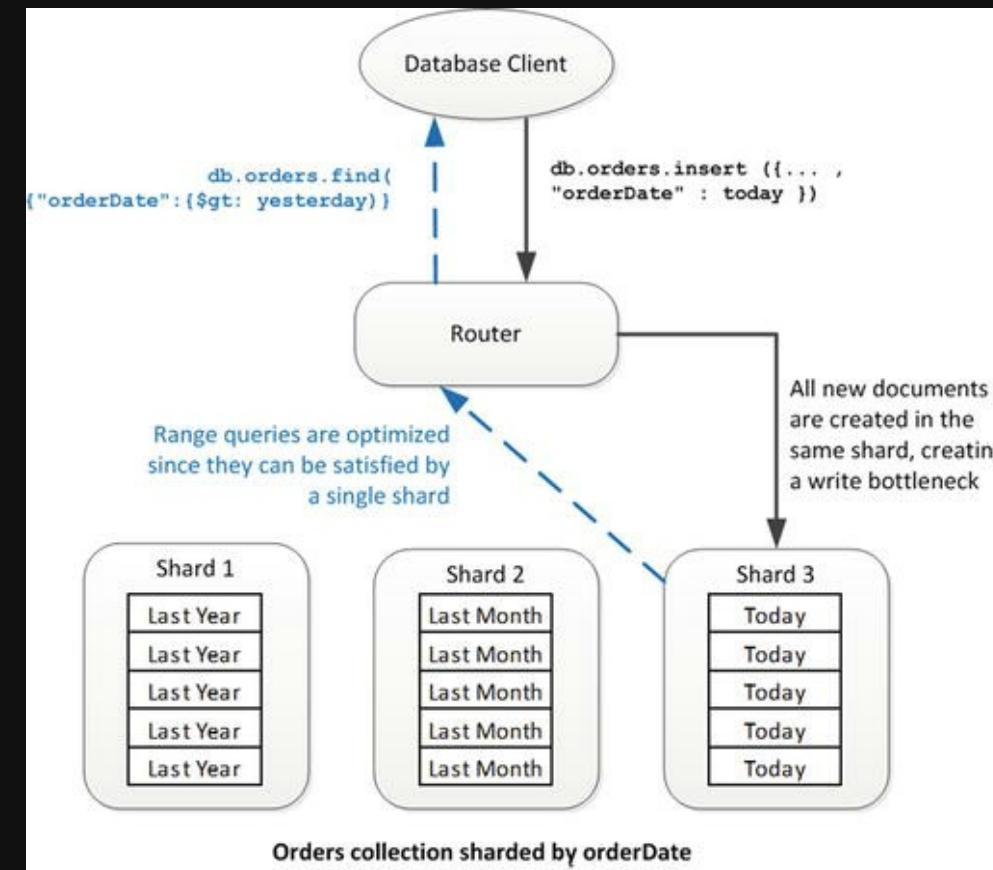
HA & Reliability

Sharding of NoSQL Database (Mongo DB)



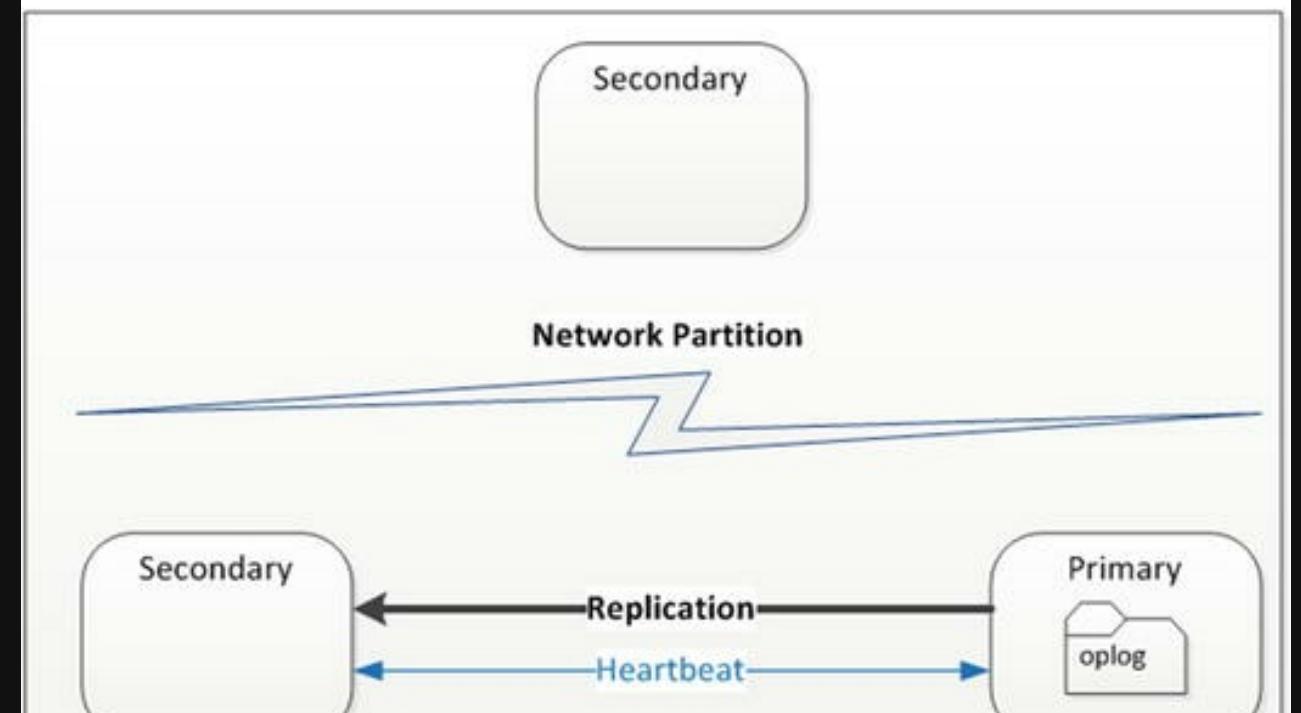
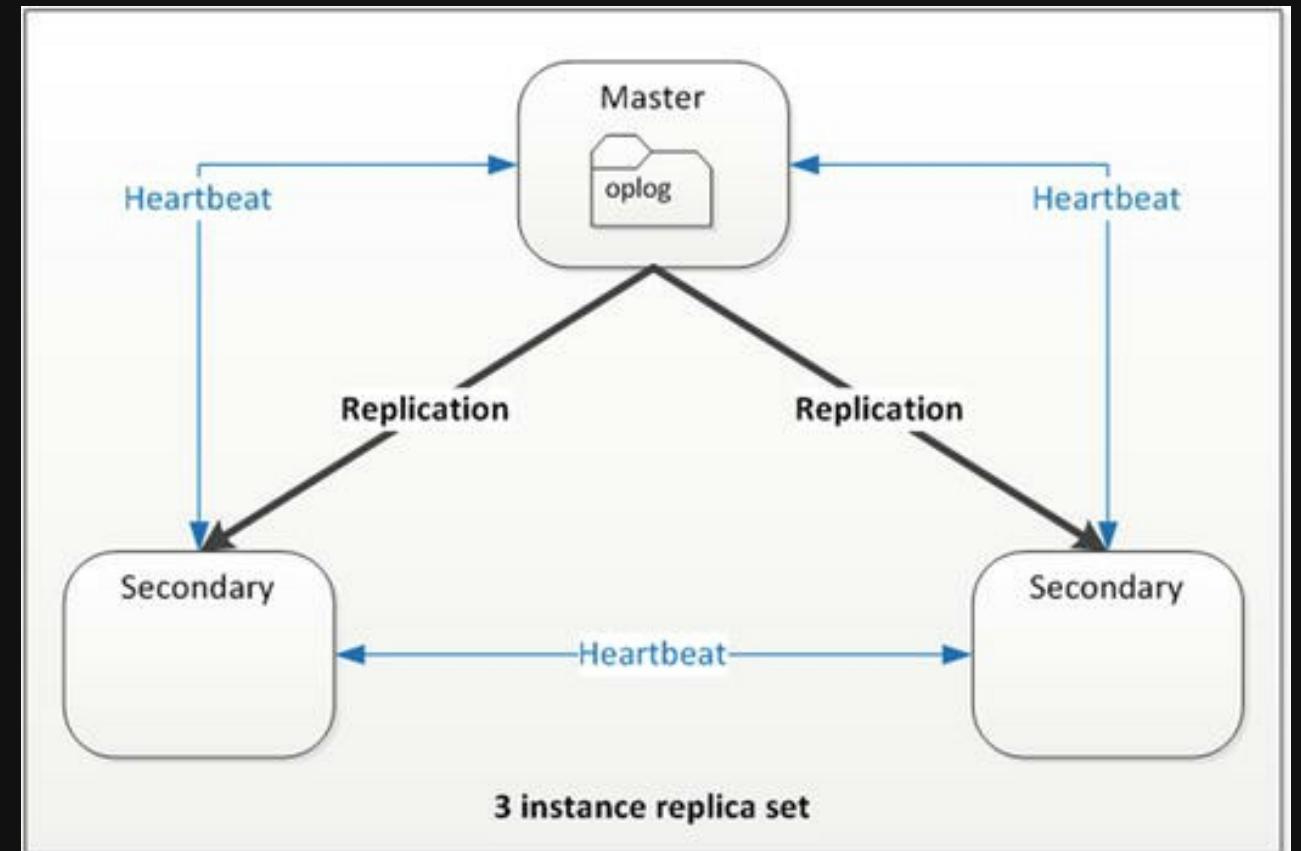
HA & Reliability

Sharding of NoSQL Database (Mongo DB .. Continued)

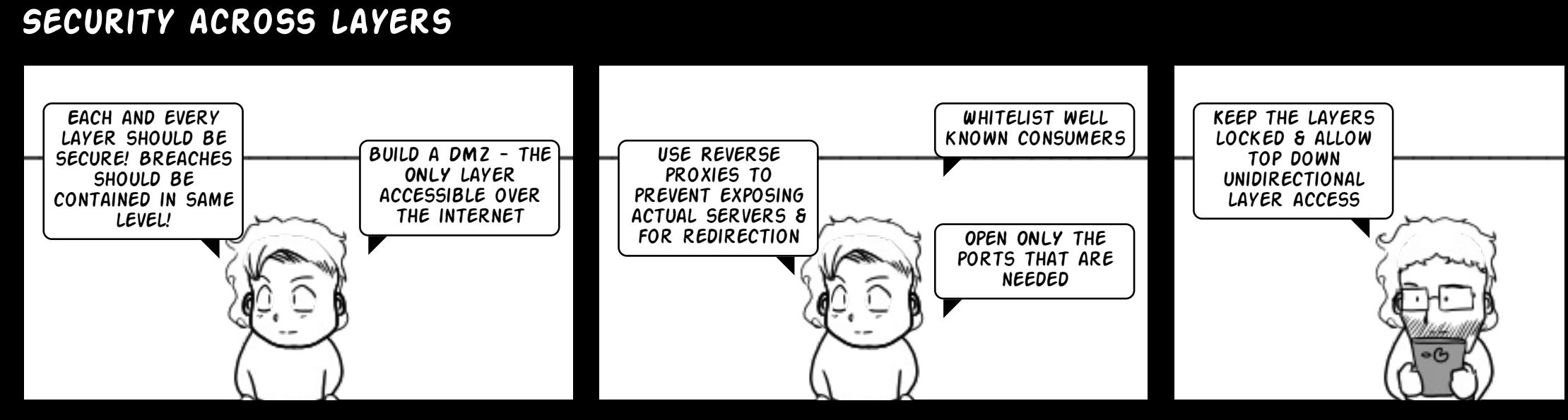


HA & Reliability

Replication set

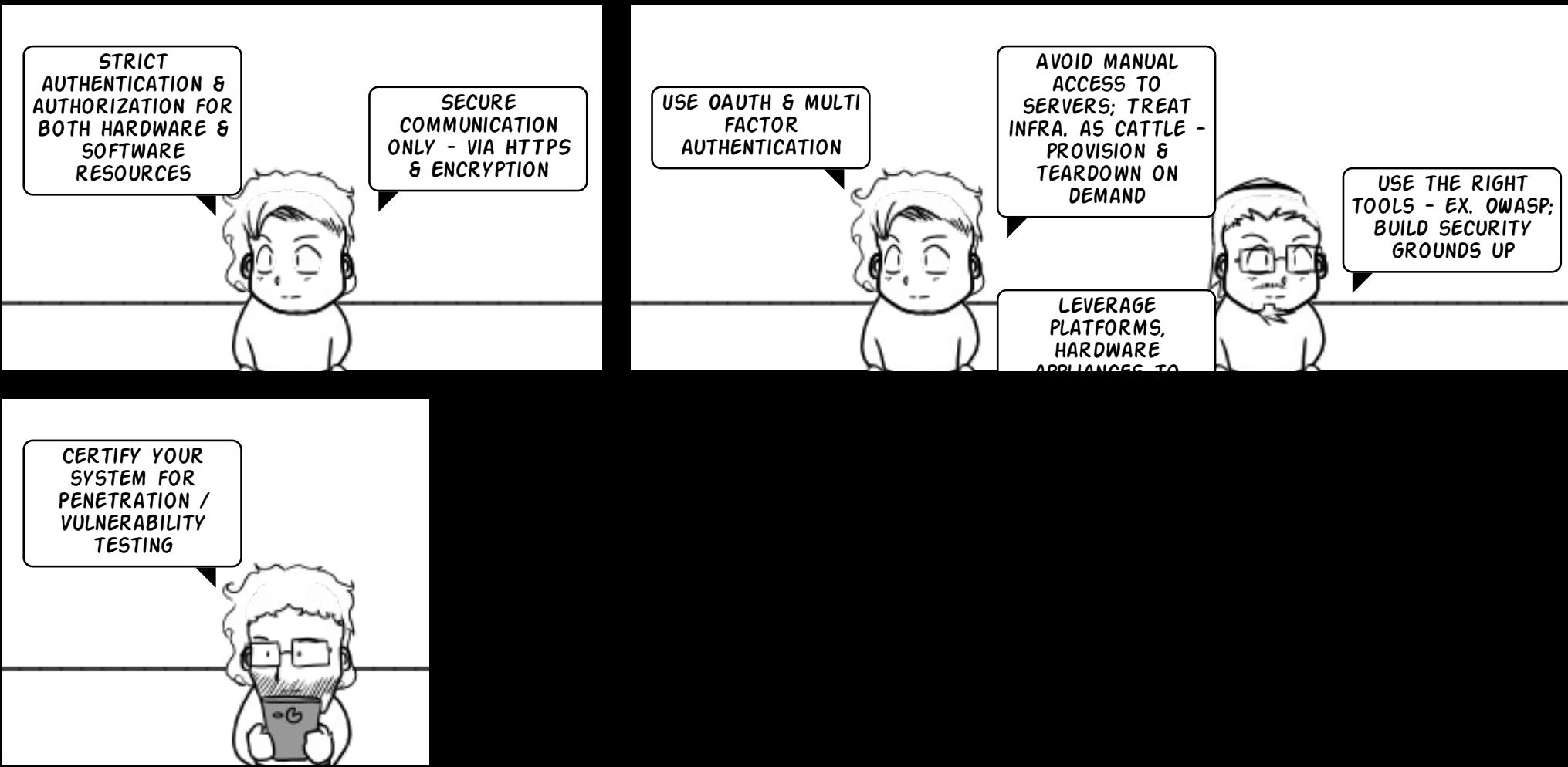


Securing the System



Securing the System

INTERNAL & EXTERNAL SECURITY

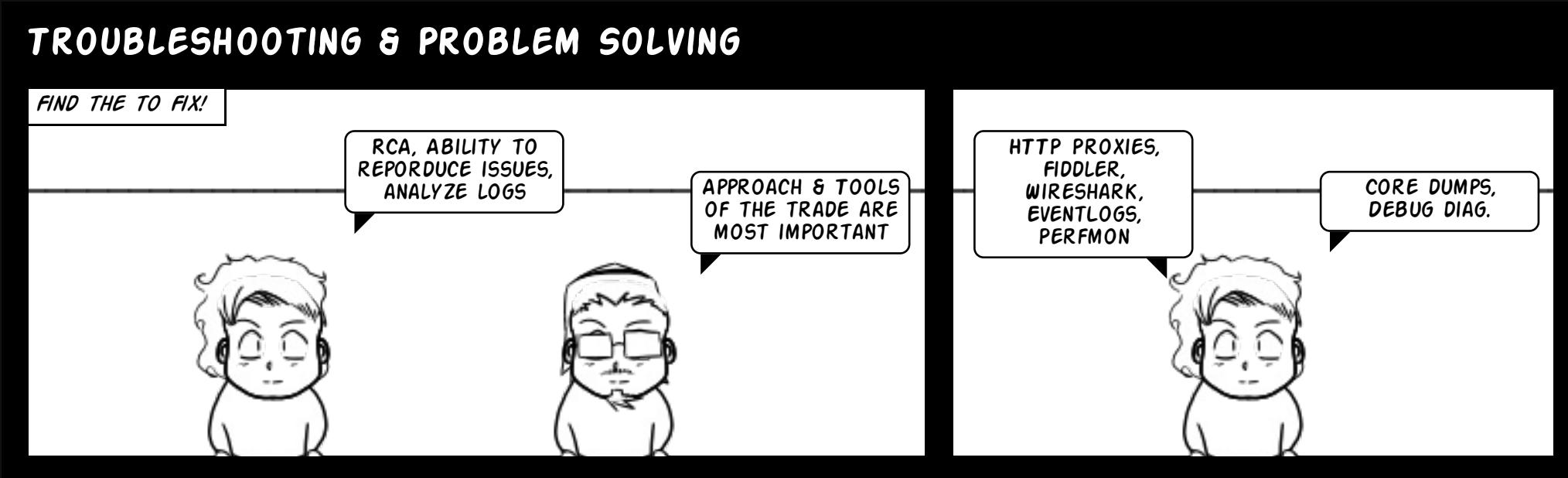


Securing the System

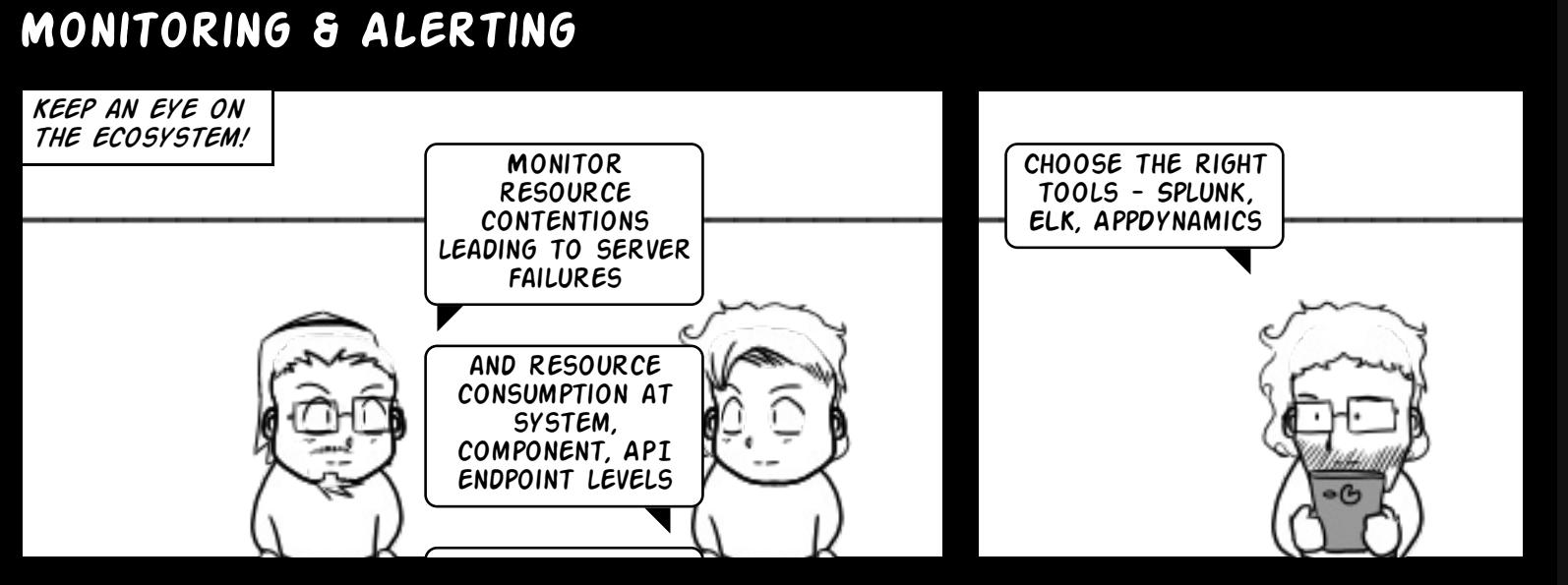
<https://www.microsoft.com/en-us/download/confirmation.aspx?id=1330>
<http://azuresecurity.codeplex.com/>

<https://msdn.microsoft.com/en-us/library/ms998530.aspx>
<https://www.microsoft.com/en-us/download/confirmation.aspx?id=11711>
<https://msdn.microsoft.com/en-us/library/ff921345.aspx>
<https://msdn.microsoft.com/en-gb/library/ff648138.aspx>
<http://dataguidance.codeplex.com/releases>

Operational needs



Operational needs



Links

Donne Martin's System Design Primer

Single absolute reference : an organized collection
of resources for system design

There's much much more, but these links give you focus

- * <http://blog.cleancoder.com/uncle-bob/2014/10/01/CleanMicroserviceArchitecture.html>
- * <http://microservices.io/>
- * <http://www.codingthearchitecture.com/>
- * <https://awesome-tech.readthedocs.io/>
- <http://alistair.cockburn.us/Hexagonal+architecture>
- <http://alistair.cockburn.us/Foundations+for+Software+Engineering>
- <http://www.idesign.net/>
- <http://www.bredemeyer.com/>
- <https://github.com/checkcheckzz/system-design-interview>
- <https://github.com/benas/awesome-software-craftsmanship>
- <https://github.com/onurakpolat/awesome-bigdata>

Books:

<https://www.manning.com/books/microservice-patterns>
<https://github.com/miguellgt/books>

THAT'S ALL FOR
NOW FOLKS!"





A journey of a thousand miles
begins with a single step
... And lots of coffee!

Visit Jim Hunt at facebook.com/huntcartoons