ARDEIM

(AR)CHITECTURE, (DE)SIGN & (IM)PLEMENTATION

THANKS TO:
RASHMI

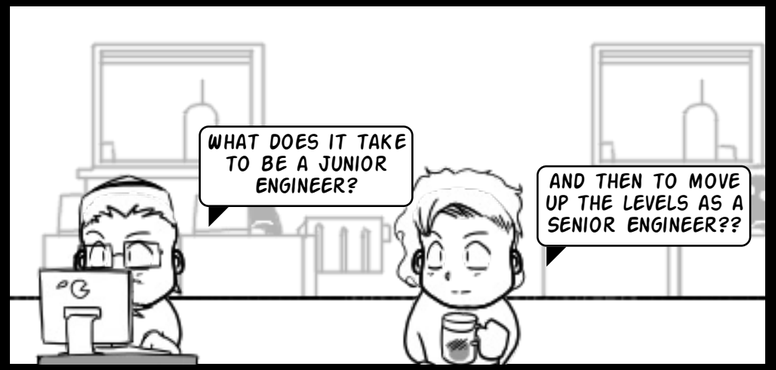
CREDITS:
REVEAL.JS, STRIPTHIS (KESIEV)

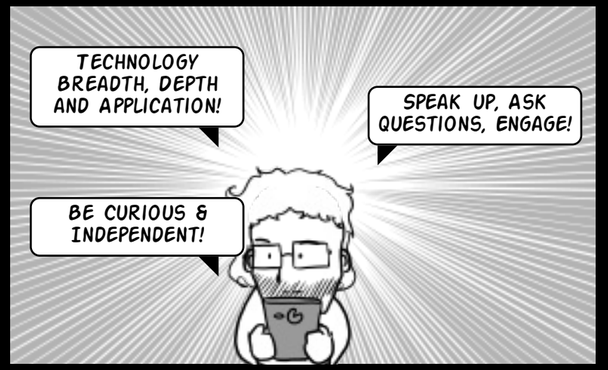
CONTEXT

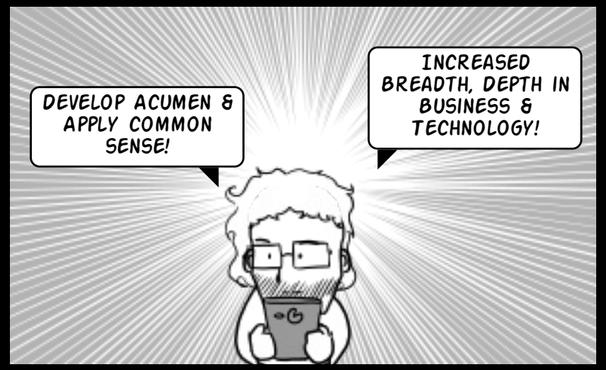

THE TEAM!
*PRODUCT MANAGER
*JR. ENGG.
*SR. ENGG.
*DIVINE VOICE!
*DEV. MANAGER

CONTEXT

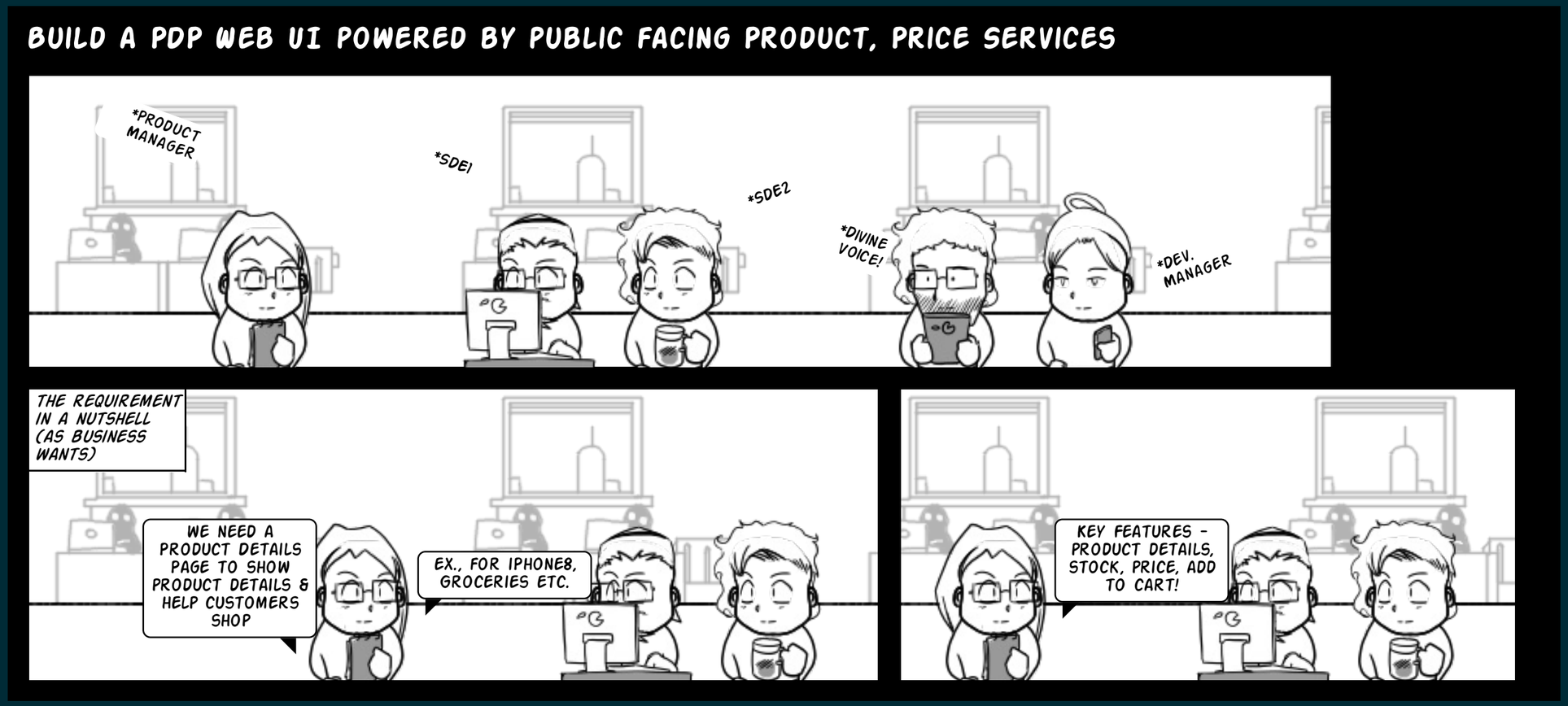
THE ENGINEER'S DILEMMA



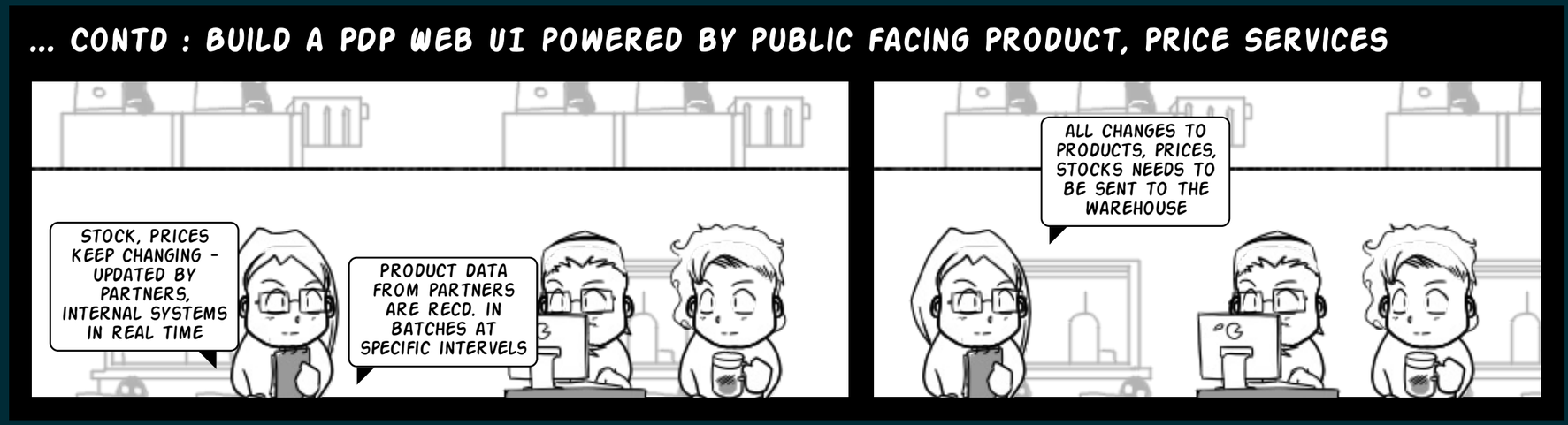




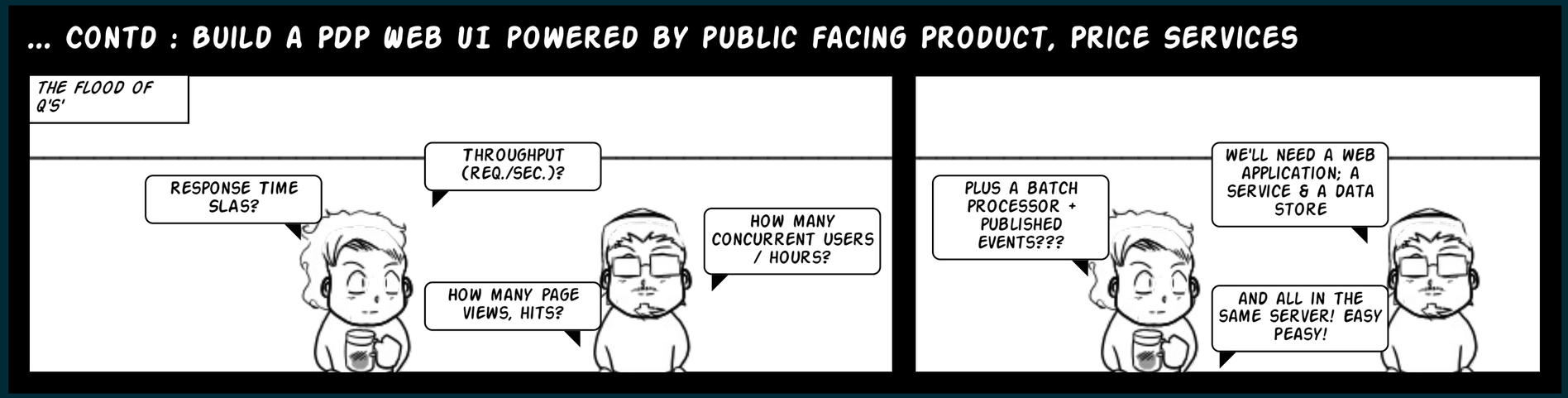
CONTEXT


BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES
*PRODUCT MANAGER
*SDE1
*SDE2
*DIVINE VOICE!
*DEV. MANAGER
THE REQUIREMENT IN A NUTSHELL (AS BUSINESS WANTS)
WE NEED A PRODUCT DETAILS PAGE TO SHOW PRODUCT DETAILS & HELP CUSTOMERS SHOP
EX., FOR IPHONE8, GROCERIES ETC.
KEY FEATURES - PRODUCT DETAILS, STOCK, PRICE, ADD TO CART!

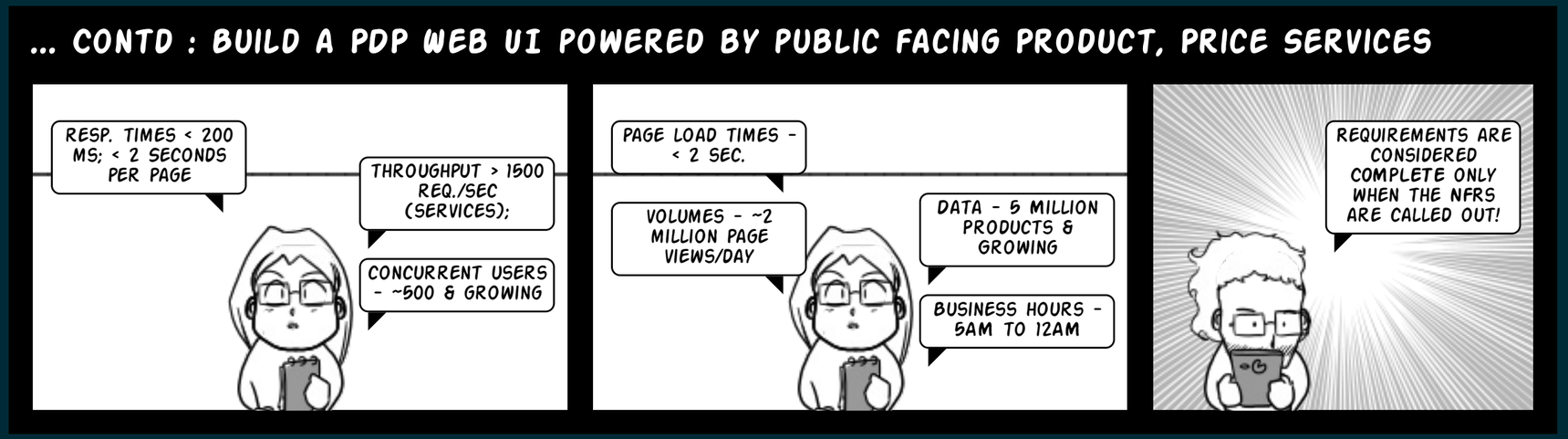
CONTEXT


... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES
STOCK, PRICES KEEP CHANGING - UPDATED BY PARTNERS, INTERNAL SYSTEMS IN REAL TIME
PRODUCT DATA FROM PARTNERS ARE RECD. IN BATCHES AT SPECIFIC INTERVELS
ALL CHANGES TO PRODUCTS, PRICES, STOCKS NEEDS TO BE SENT TO THE WAREHOUSE

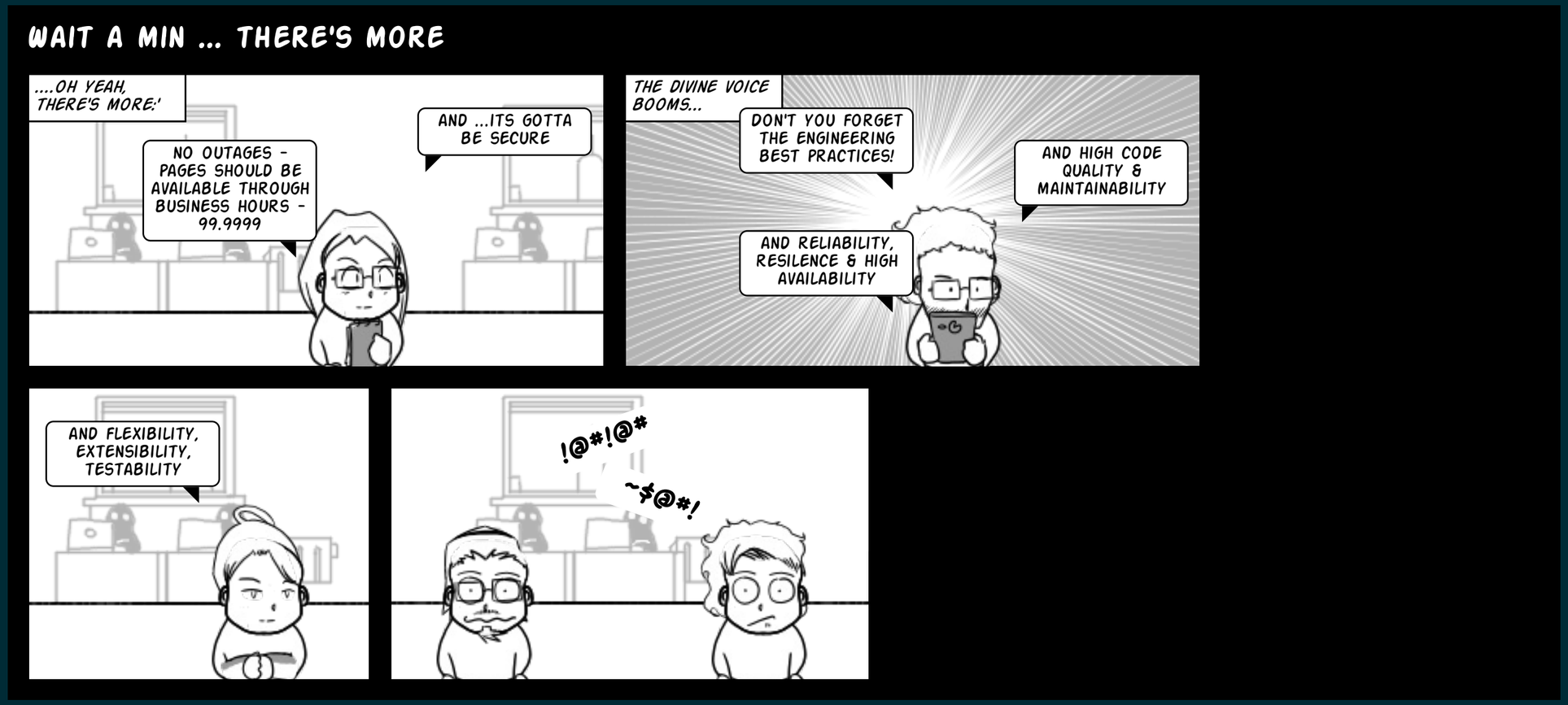
CONTEXT


... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES
THE FLOOD OF Q'S'
RESPONSE TIME SLAS?
THROUGHPUT (REQ./SEC.)?
HOW MANY PAGE VIEWS, HITS?
HOW MANY CONCURRENT USERS / HOURS?
PLUS A BATCH PROCESSOR + PUBLISHED EVENTS???
WE'LL NEED A WEB APPLICATION; A SERVICE & A DATA STORE
AND ALL IN THE SAME SERVER! EASY PEASY!

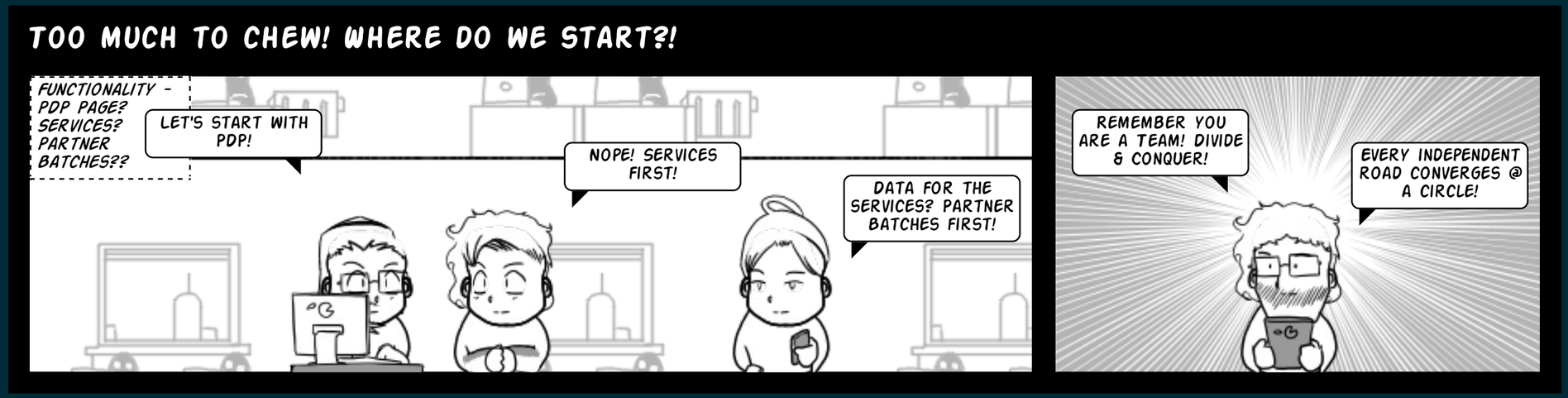
CONTEXT


... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES
RESP. TIMES < 200 MS; < 2 SECONDS PER PAGE
THROUGHPUT > 1500 REQ./SEC (SERVICES);
CONCURRENT USERS - ~500 & GROWING
PAGE LOAD TIMES - < 2 SEC.
VOLUMES - ~2 MILLION PAGE VIEWS/DAY
DATA - 5 MILLION PRODUCTS & GROWING
BUSINESS HOURS - 5AM TO 12AM
REQUIREMENTS ARE CONSIDERED COMPLETE ONLY WHEN THE NFRS ARE CALLED OUT!

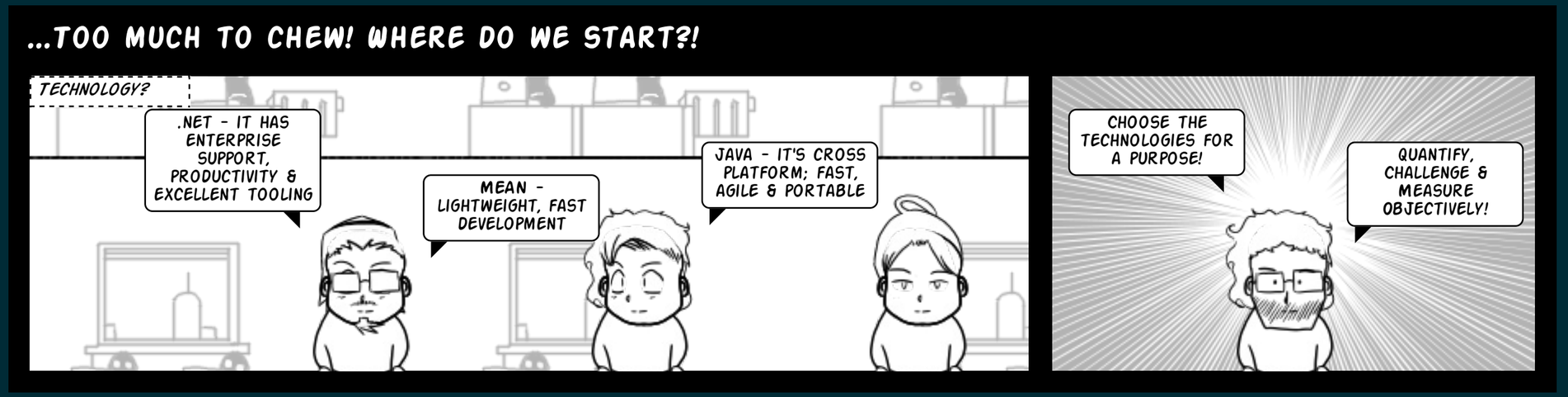
CONTEXT


WAIT A MIN ... THERE'S MORE
....OH YEAH, THERE'S MORE:'
NO OUTAGES - PAGES SHOULD BE AVAILABLE THROUGH BUSINESS HOURS - 99.9999
AND ...ITS GOTTA BE SECURE
THE DIVINE VOICE BOOMS...
DON'T YOU FORGET THE ENGINEERING BEST PRACTICES!
AND HIGH CODE QUALITY & MAINTAINABILITY
AND RELIABILITY, RESILENCE & HIGH AVAILABILITY
AND FLEXIBILITY, EXTENSIBILITY, TESTABILITY
!@*!@*
~*@*!

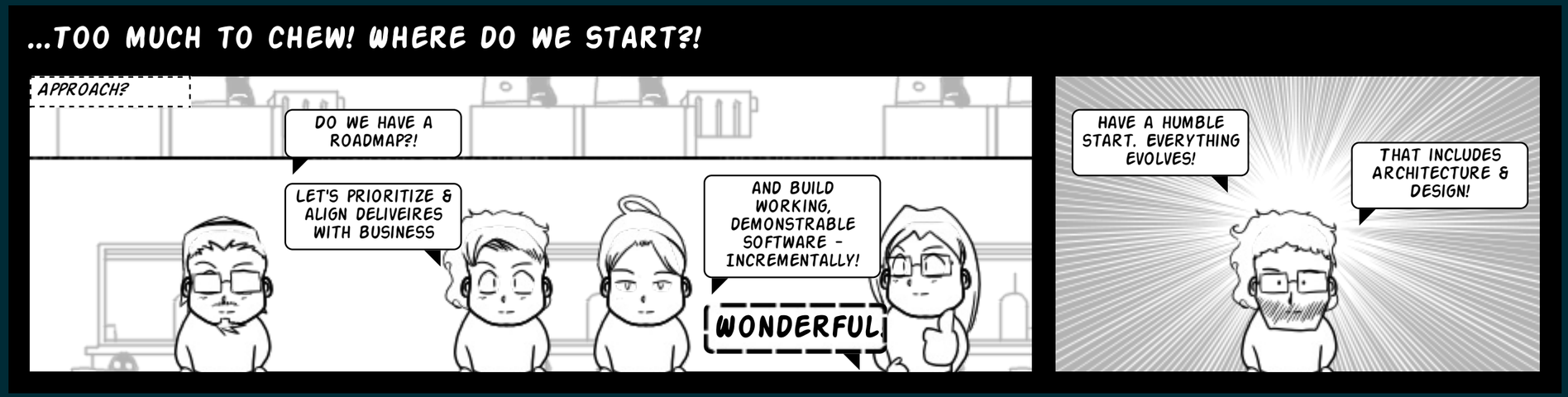
WHAT NEXT!?


TOO MUCH TO CHEW! WHERE DO WE START?!
FUNCTIONALITY - PDP PAGE? SERVICES? PARTNER BATCHES??
LET'S START WITH PDP!
NOPE! SERVICES FIRST!
DATA FOR THE SERVICES? PARTNER BATCHES FIRST!
REMEMBER YOU ARE A TEAM! DIVIDE & CONQUER!
EVERY INDEPENDENT ROAD CONVERGES @ A CIRCLE!

WHAT NEXT!?


...TOO MUCH TO CHEW! WHERE DO WE START?!
TECHNOLOGY?
.NET - IT HAS ENTERPRISE SUPPORT, PRODUCTIVITY & EXCELLENT TOOLING
MEAN - LIGHTWEIGHT, FAST DEVELOPMENT
JAVA - IT'S CROSS PLATFORM; FAST, AGILE & PORTABLE
CHOOSE THE TECHNOLOGIES FOR A PURPOSE!
QUANTIFY, CHALLENGE & MEASURE OBJECTIVELY!

WHAT NEXT!?


...TOO MUCH TO CHEW! WHERE DO WE START?!
APPROACH?
DO WE HAVE A ROADMAP?!
LET'S PRIORITIZE & ALIGN DELIVEIRES WITH BUSINESS
AND BUILD WORKING, DEMONSTRABLE SOFTWARE - INCREMENTALLY!
WONDERFUL
HAVE A HUMBLE START. EVERYTHING EVOLVES!
THAT INCLUDES ARCHITECTURE & DESIGN!

WHAT NEXT!?

Links: Domain Driven Design - Eric Evans

WHAT NEXT!?

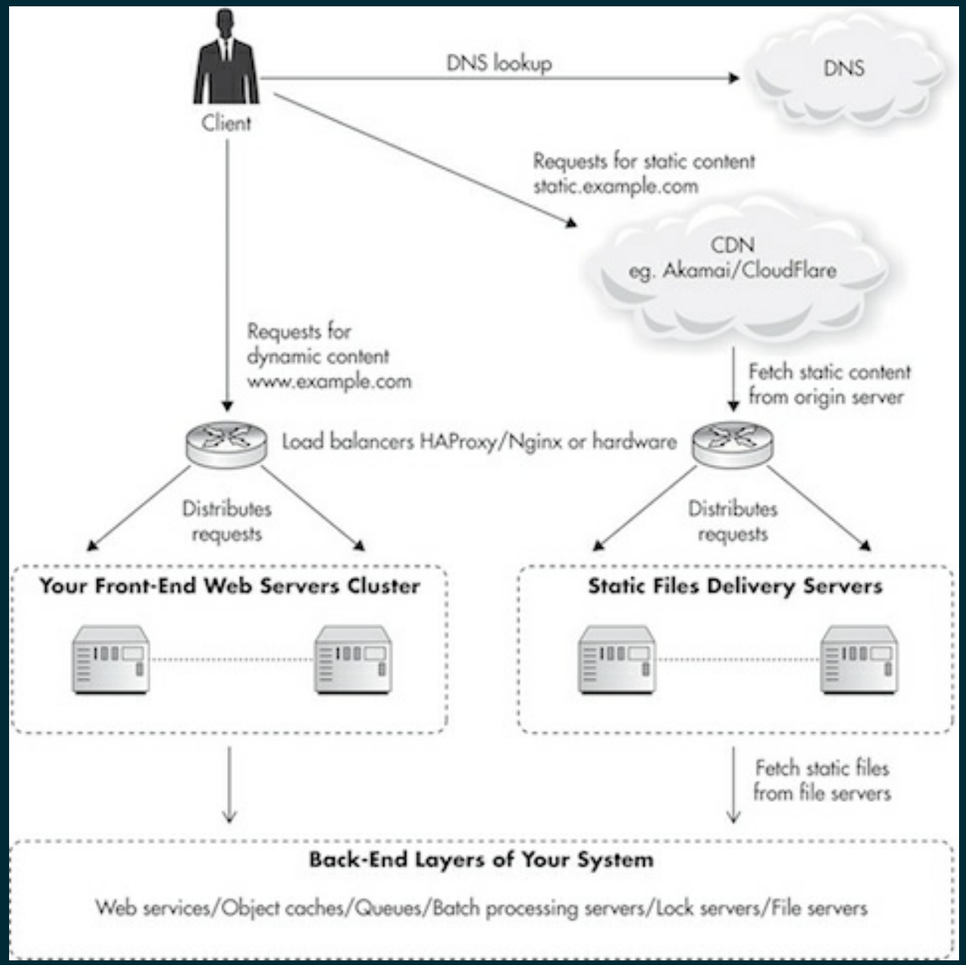

...TOO MUCH TO CHEW! WHERE DO WE START?!
WHAT'D BE THE PRO'S & CON'S? LET'S PICK THE BEST TECH. STACK SUITED TO OUR NEEDS.
GOOD GRASP OF AT LEAST ONE MAJOR PROGRAMMING LANGUAGE
A POLYGLOT, PREFERABLY
DEVELOP BREADTH, KEEP UPDATED ABOUT INDUSTRY NEWS & DEVELOPMENTS

WHAT NEXT!?


...CONTD!
JAVA VS .NET VS .NET CORE VS MEAN???
SPRING OR WEBAPIS??
VERT.X OR NODE??
PLAY OR NANCY?
RXJAVA OR AKKA??
DDD & DOMAIN MODELLING FOR SERVICES

WHAT NEXT!?

Which Technology to choose for App Layer?



NGINX beased on event loop scales well also provides load balncing and reverse proxy capabilities

WHAT NEXT!?

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

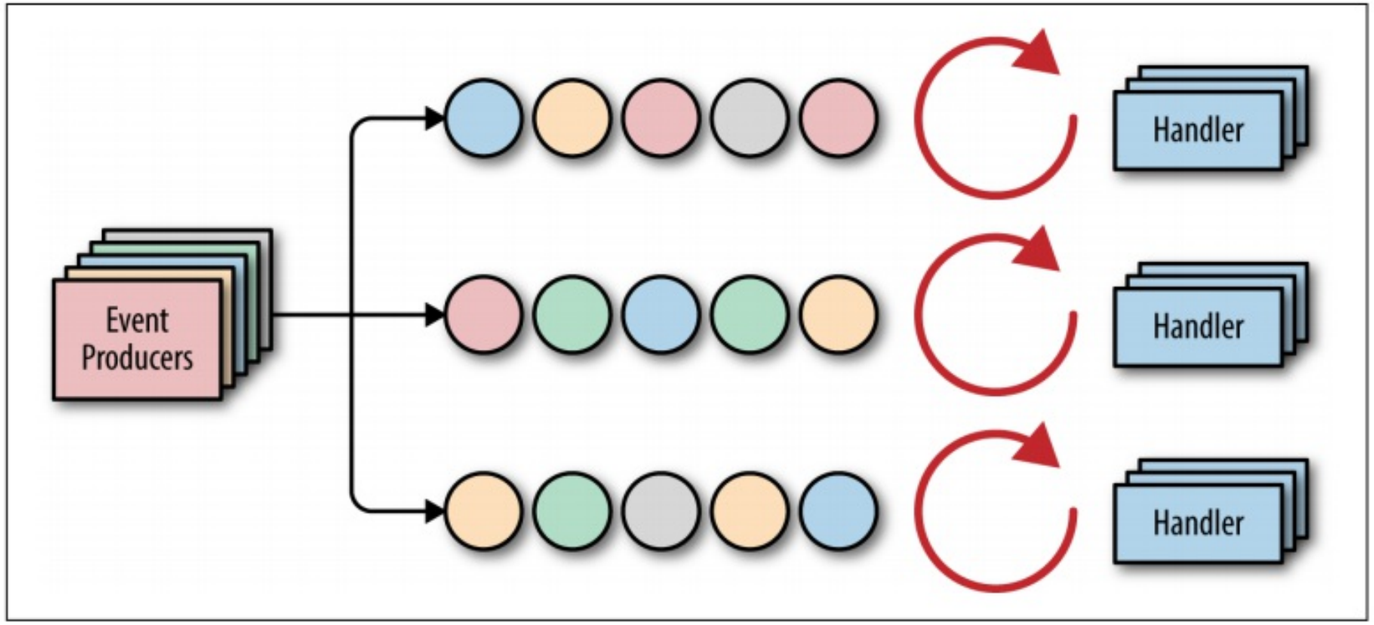
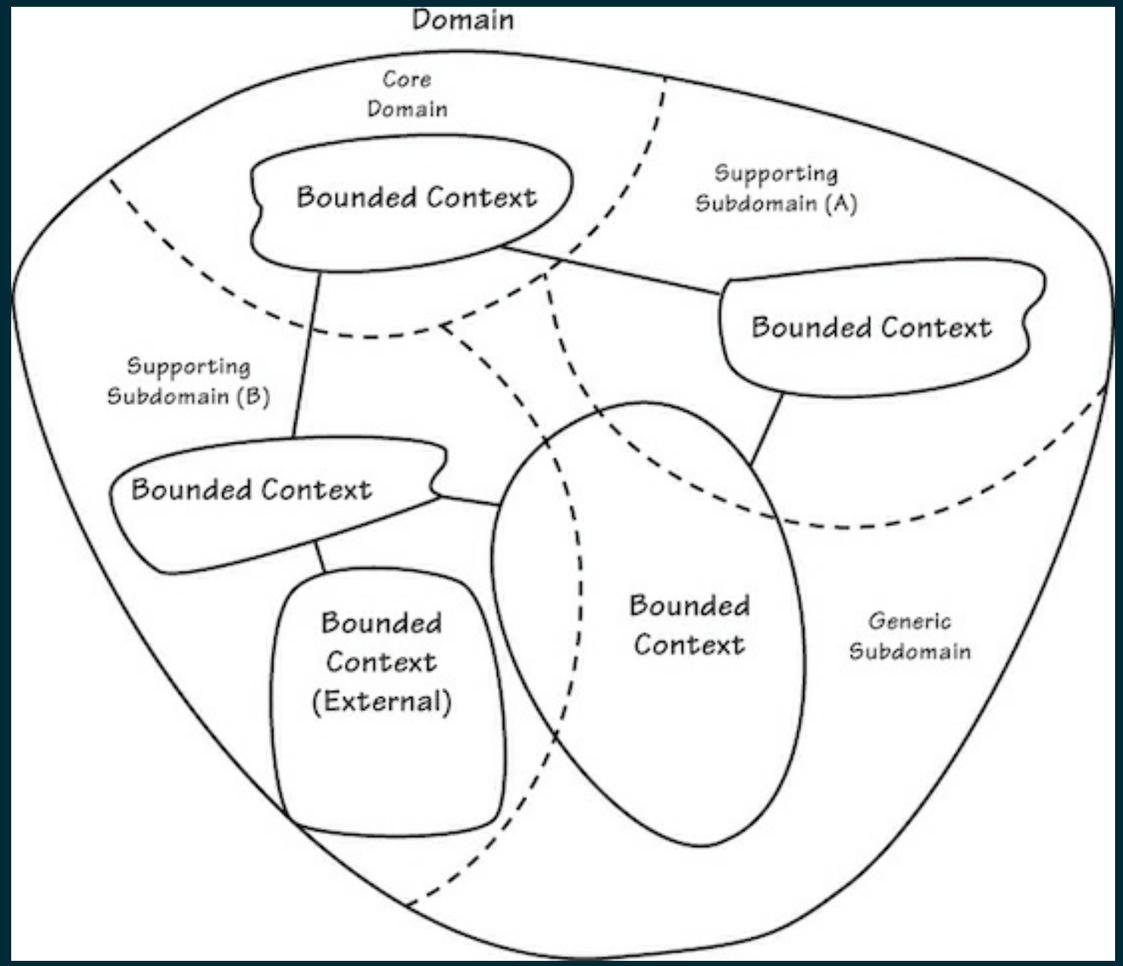
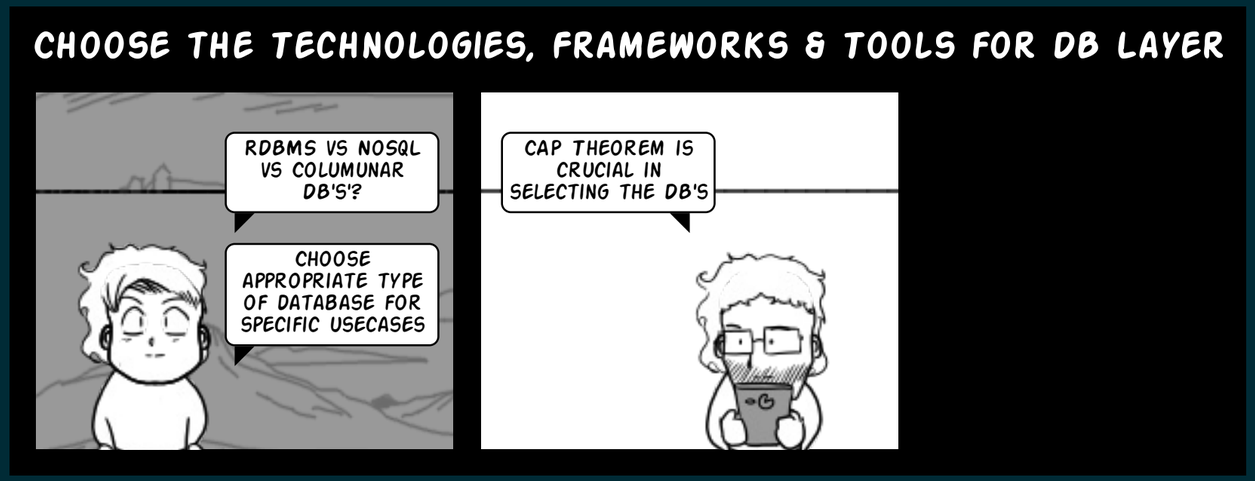


Figure 2-4. The multireactor principle

WHAT NEXT!?


Domain
Core
Domain
Bounded Context
Supporting
Subdomain (A)
Bounded Context
Supporting
Subdomain (B)
Bounded Context
Bounded
Context
(External)
Bounded
Context
Generic
Subdomain

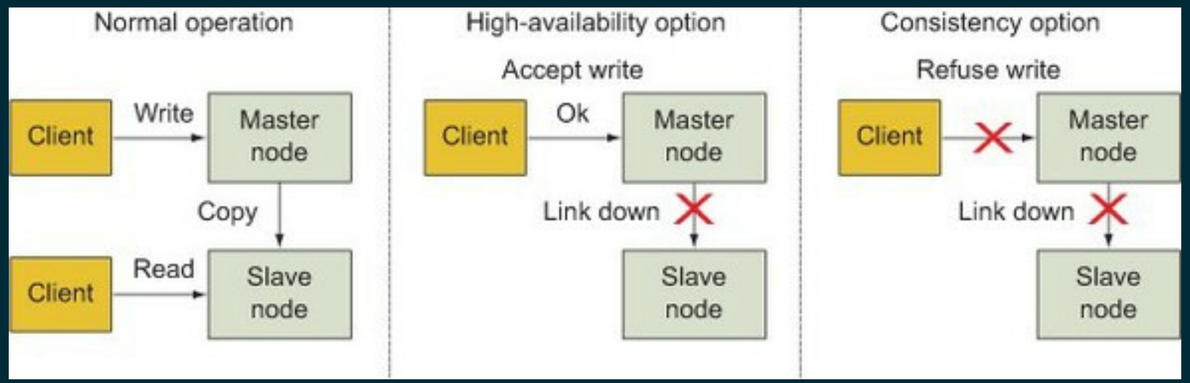
WHAT NEXT!?


CHOOSE THE TECHNOLOGIES, FRAMEWORKS & TOOLS FOR DB LAYER
RDBMS VS NOSQL VS COLUMUNAR DB'S'?
CHOOSE APPROPRIATE TYPE OF DATABASE FOR SPECIFIC USECASES
CAP THEOREM IS CRUCIAL IN SELECTING THE DB'S

WHAT NEXT!?

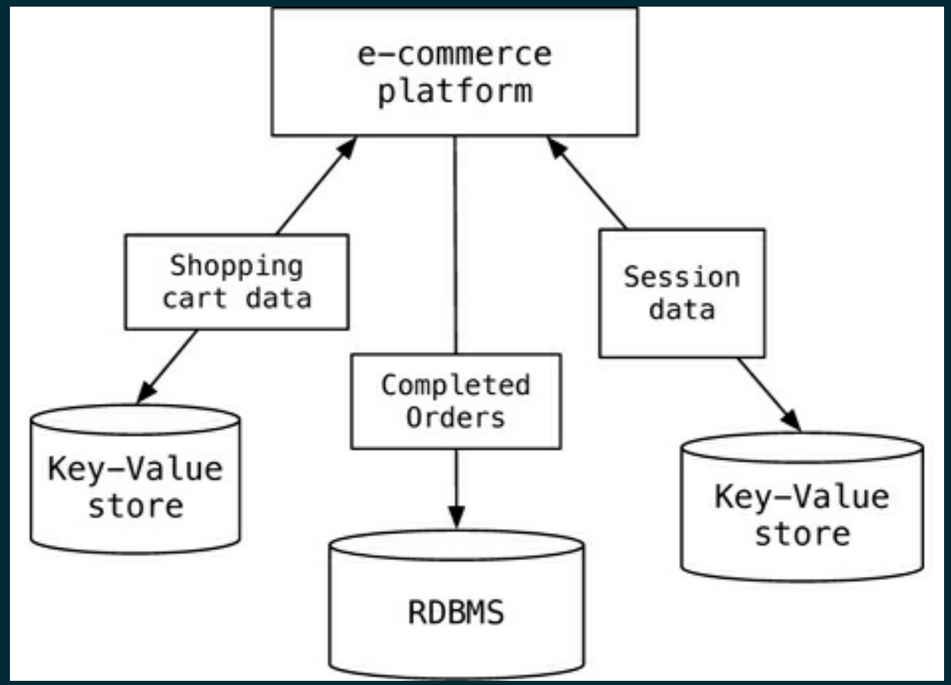
CAP Therom

Availability , Constitancy, Partition Tolerance



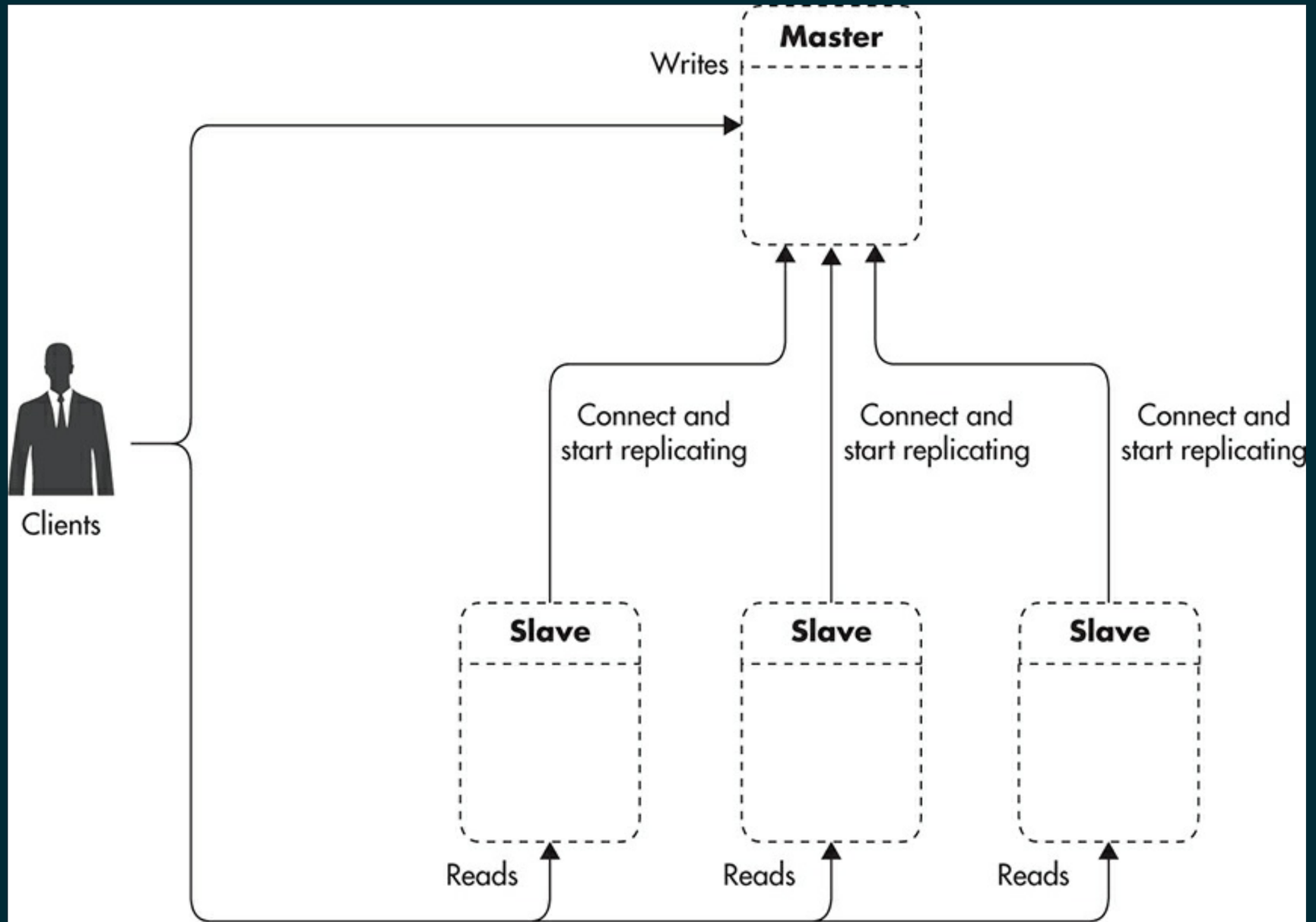
WHAT NEXT!?

Ploygot DataBase for different purpose



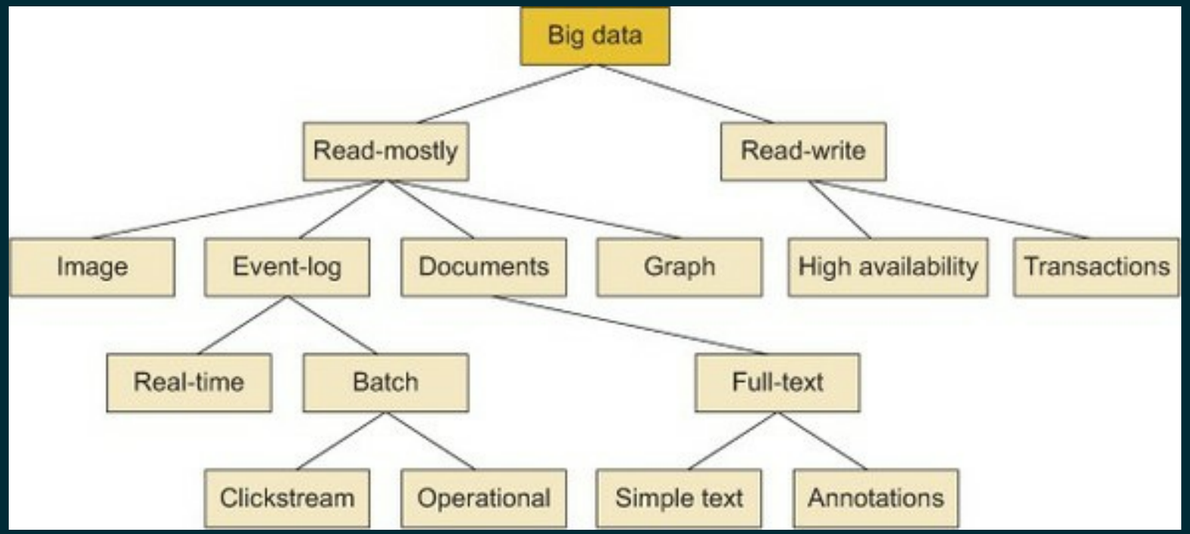
WHAT NEXT!?

Ploygot DataBase for different purpose



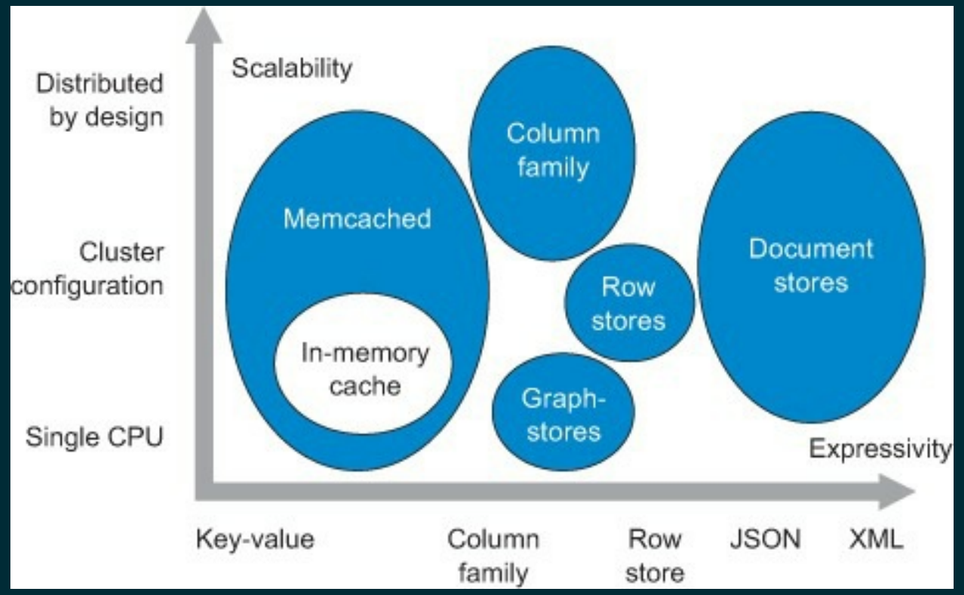
WHAT NEXT!?

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

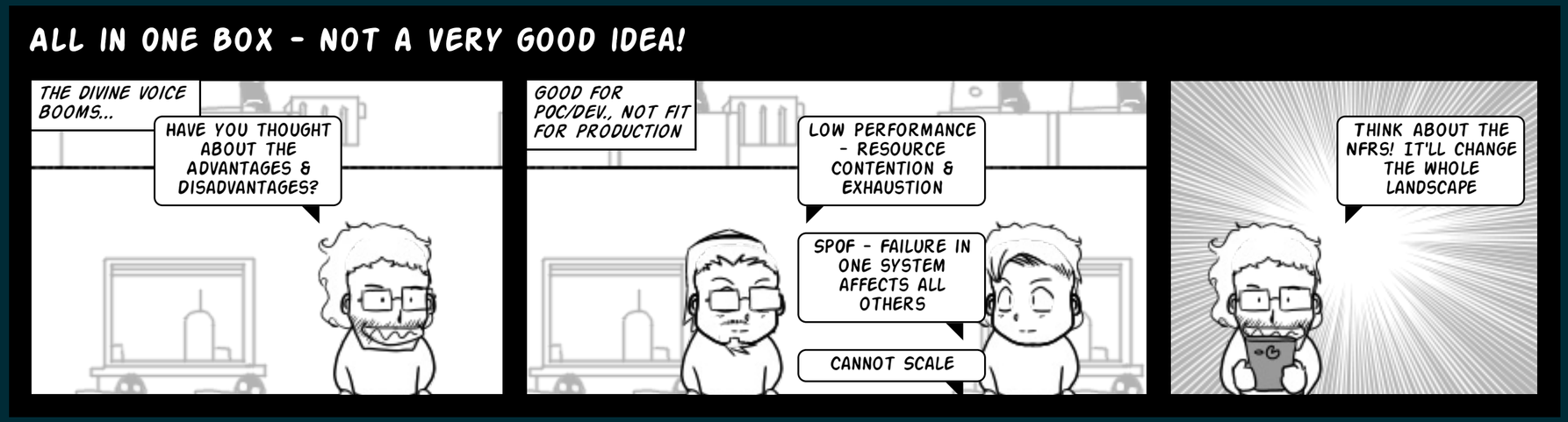


WHAT NEXT!?

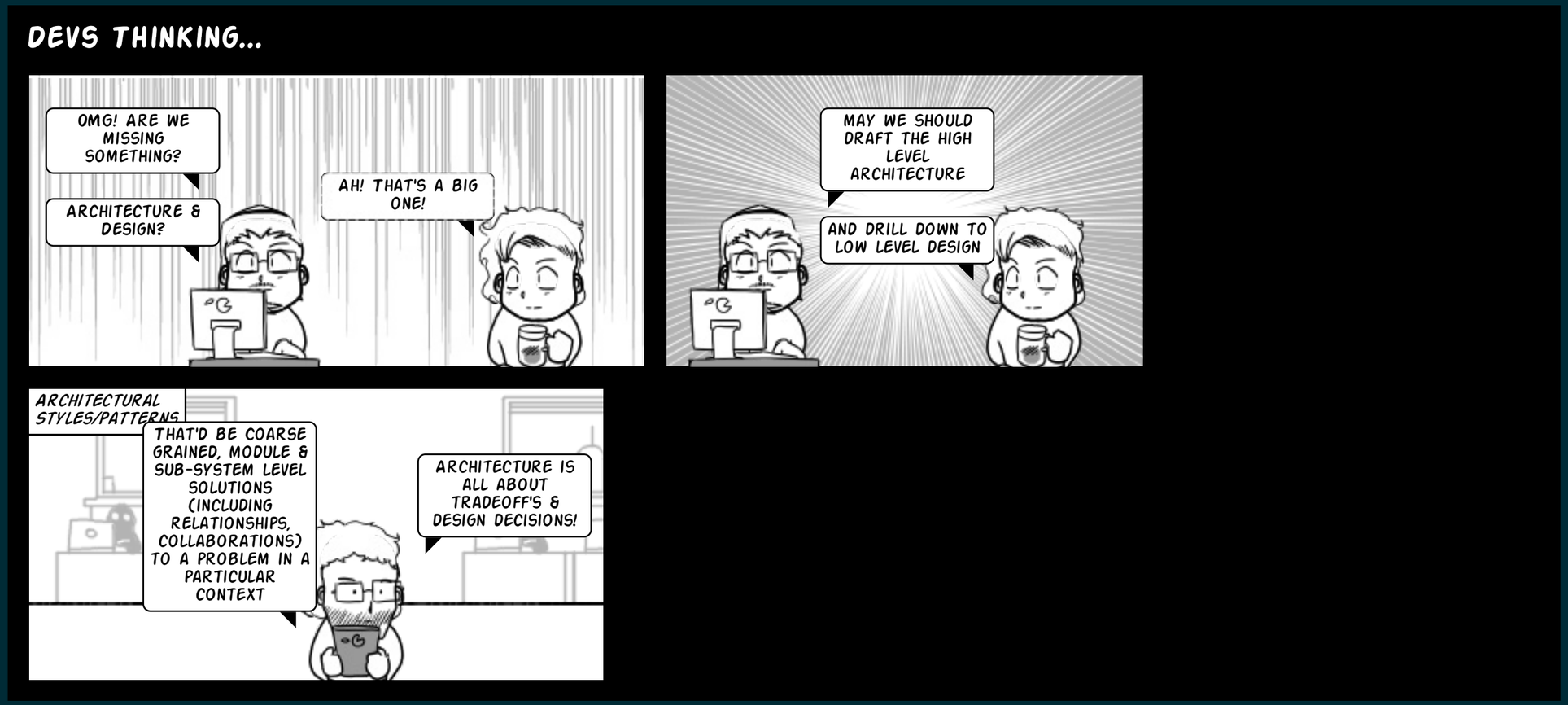
Which Technology to choose for nature of data
Choose the right data base based on the type of data



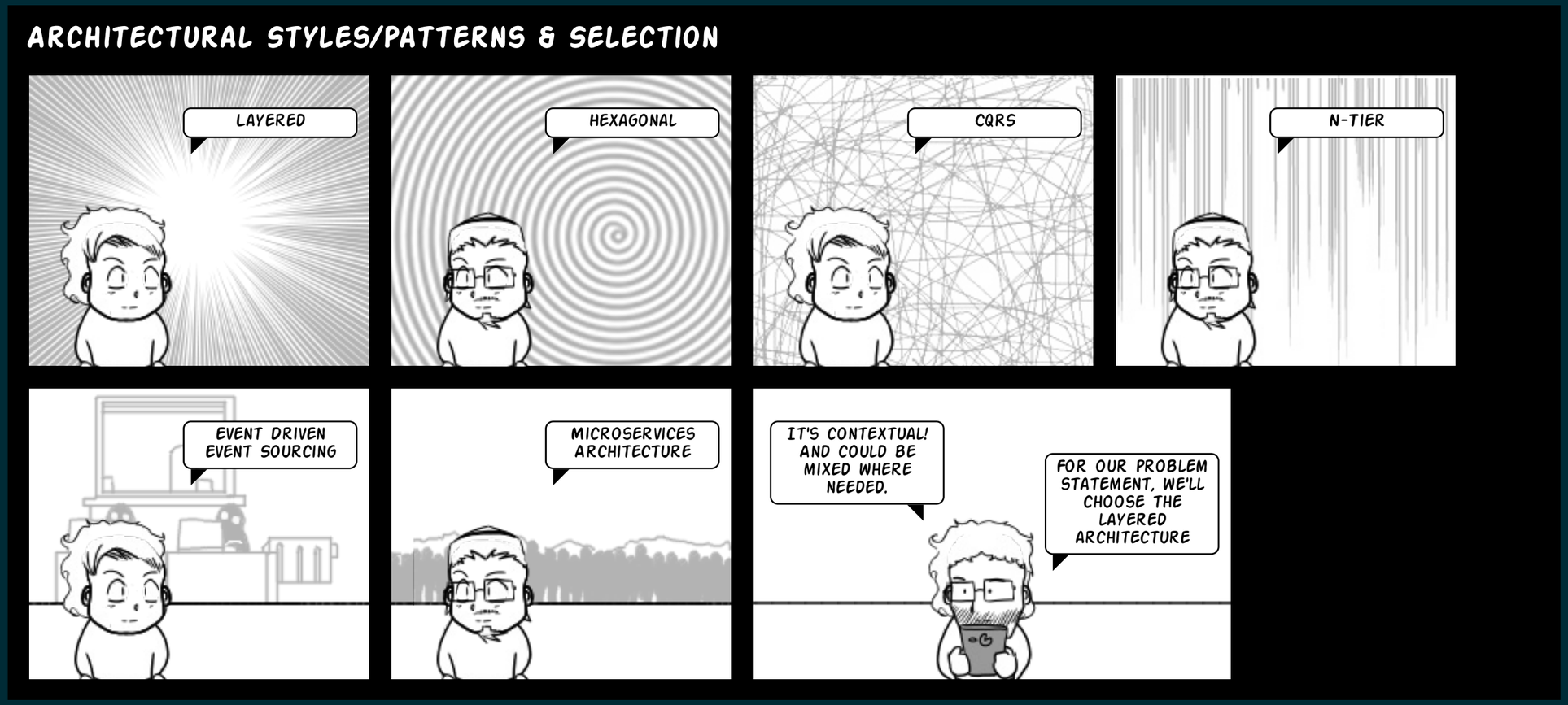
ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...

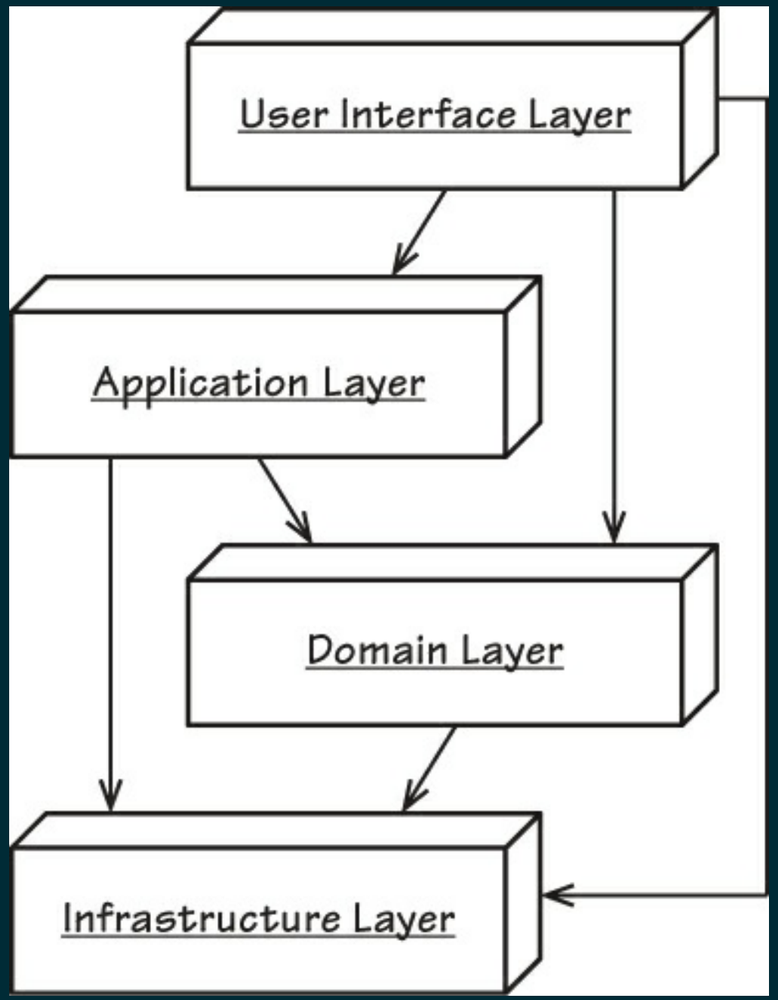


ARCHITECTURE & DESIGN ...

* [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 ...

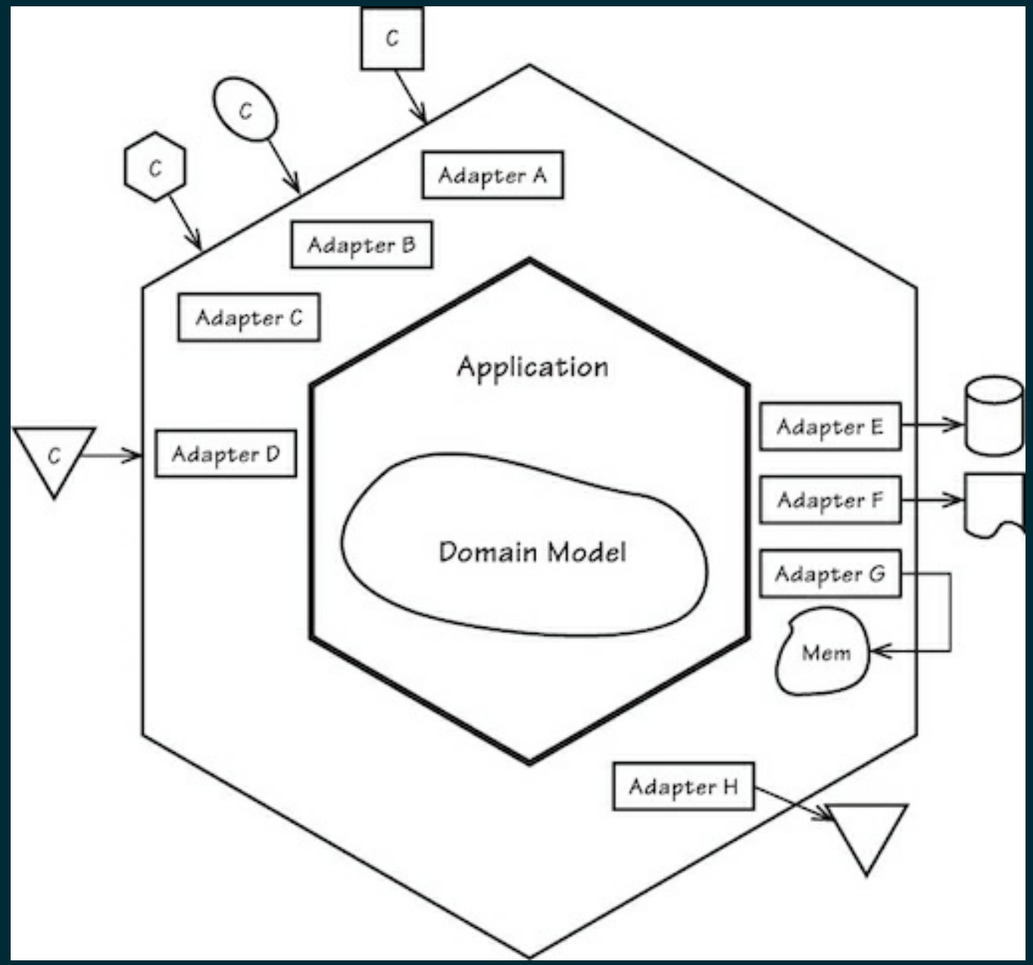
Layered Architecture



First Basic Architecture

ARCHITECTURE & DESIGN ...

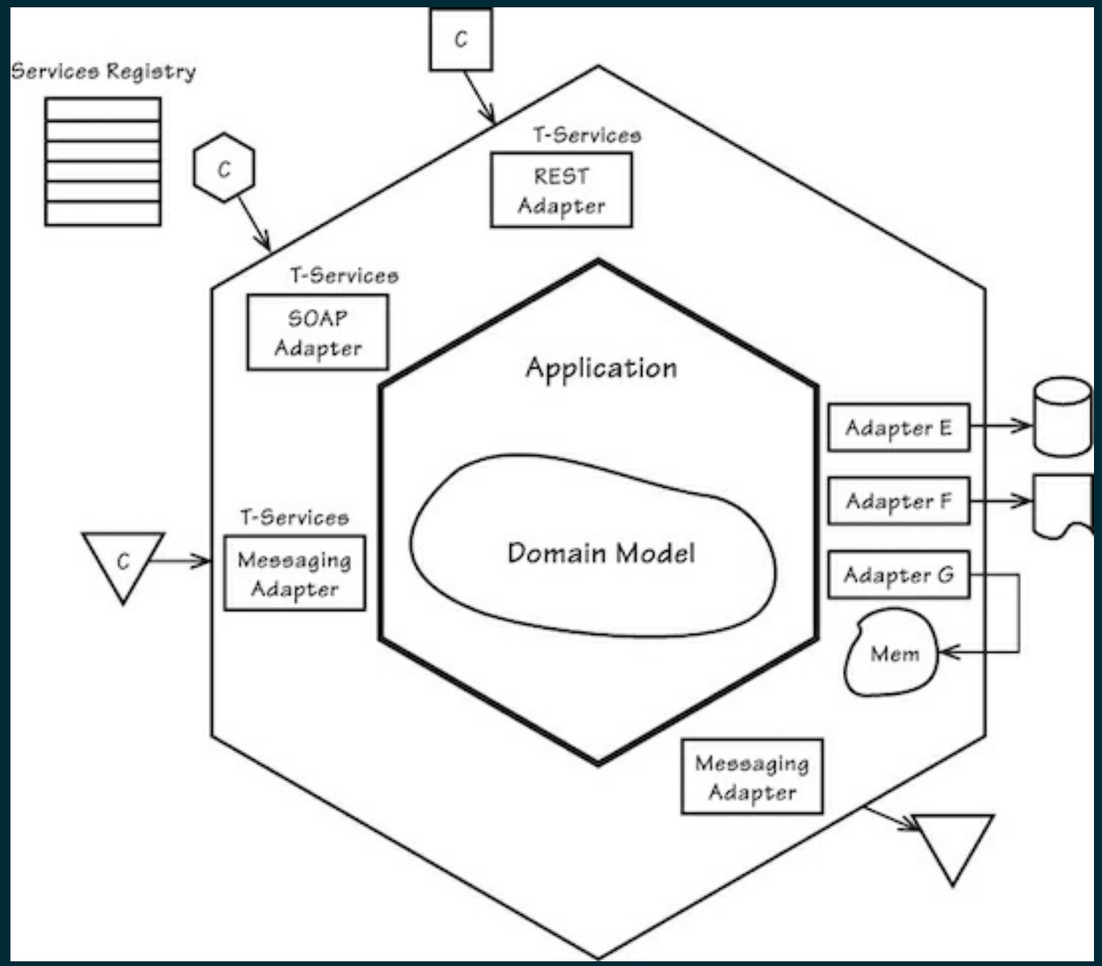
Hexagonal Architecture



The concepts of Adapters with CoreDomain

ARCHITECTURE & DESIGN ...

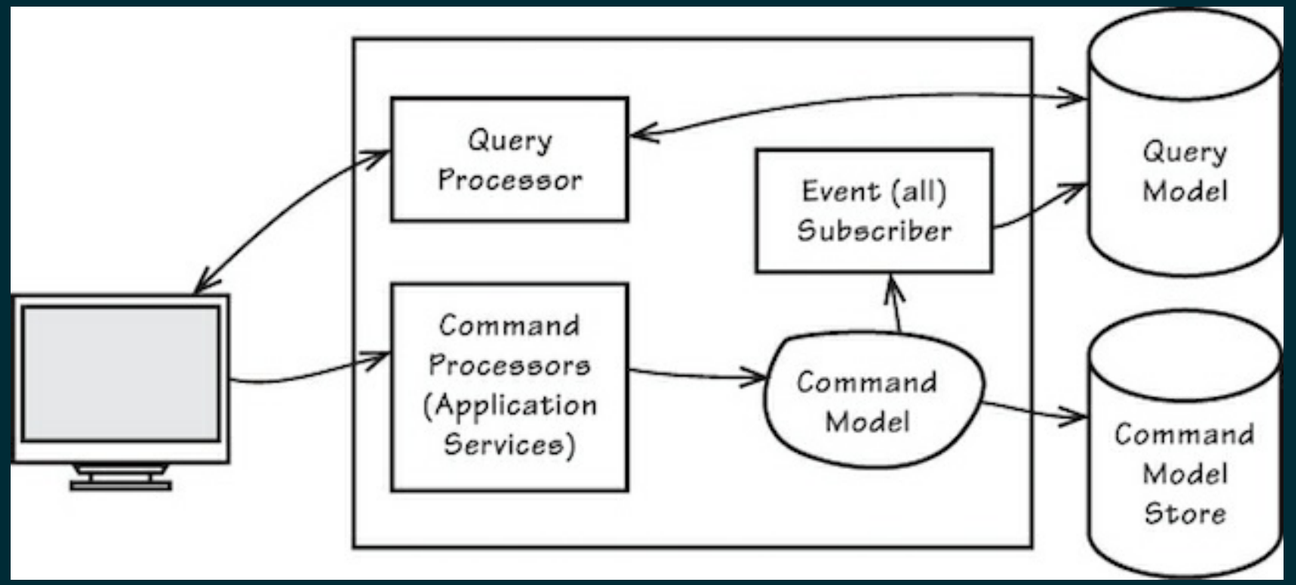
Hexogonal Architecture SOA



The concepts of Adapters with CoreDomain & with SOA

ARCHITECTURE & DESIGN ...

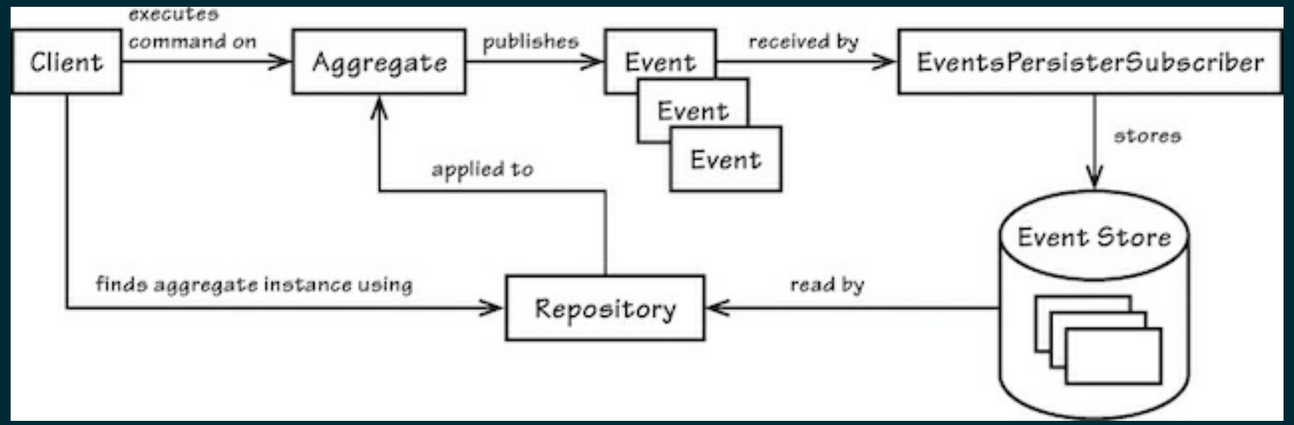
CQRS



Seperate FLow for Query and Create

ARCHITECTURE & DESIGN ...

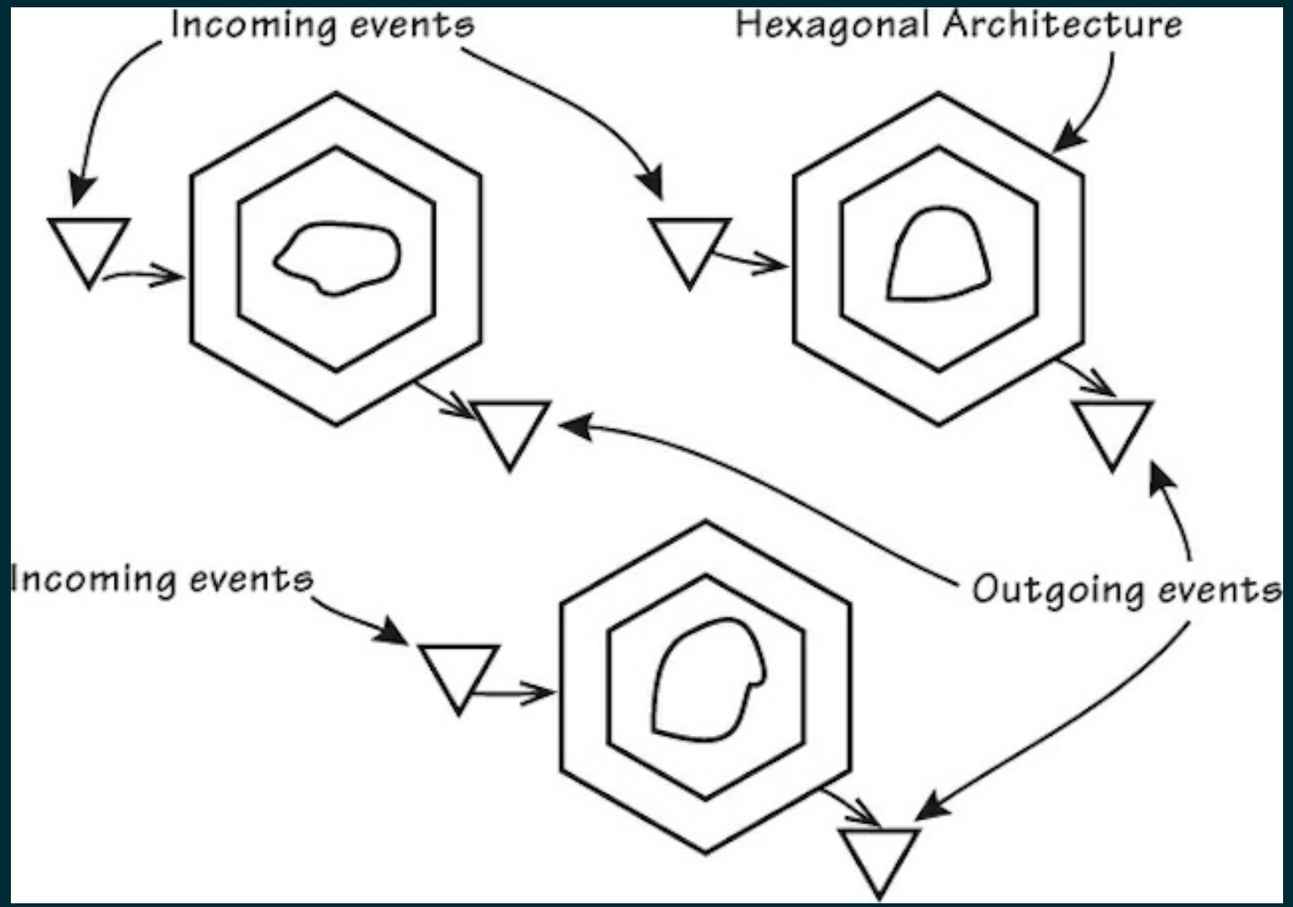
Event Sourcing



Events are stored no events are updates or deleted

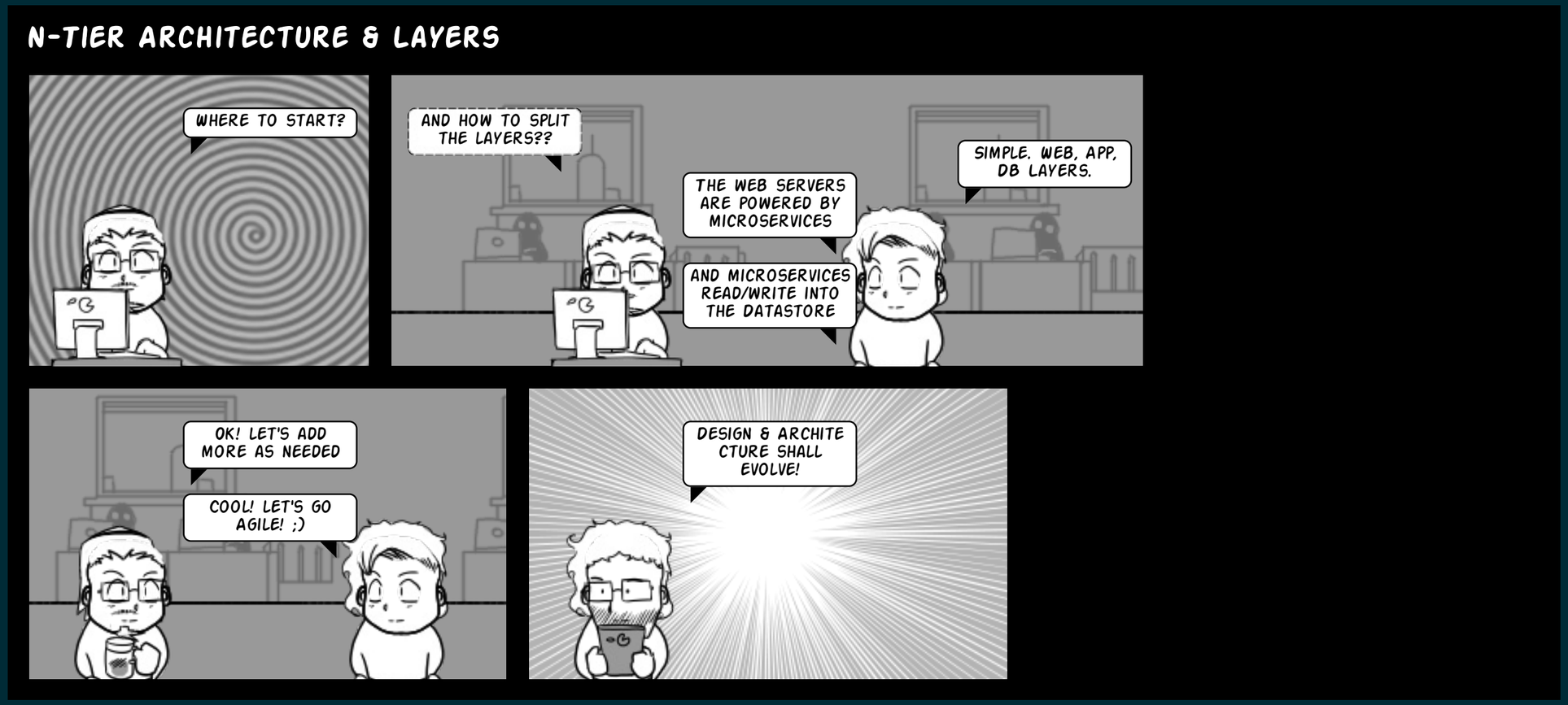
ARCHITECTURE & DESIGN ...

Event Driven

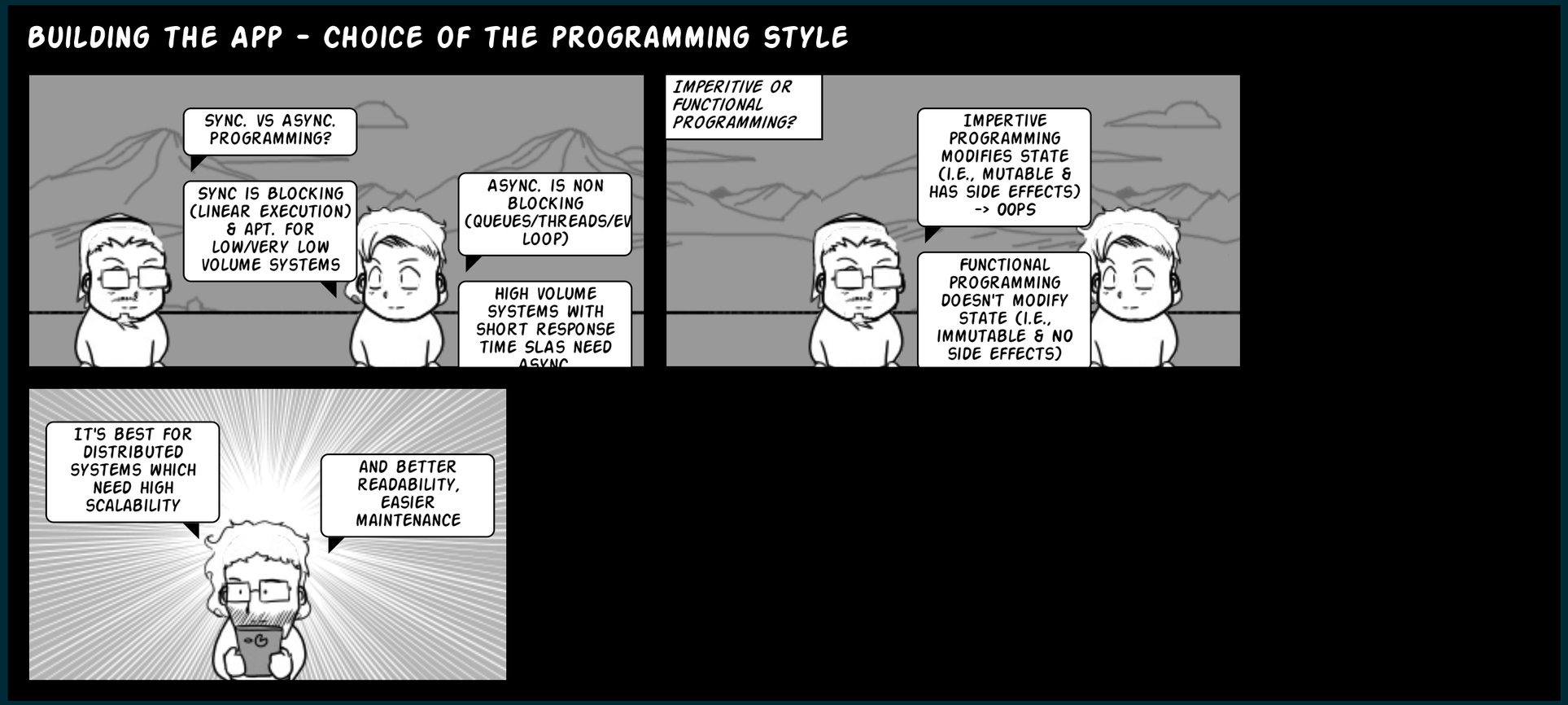


Different System integrtate using Domain Events

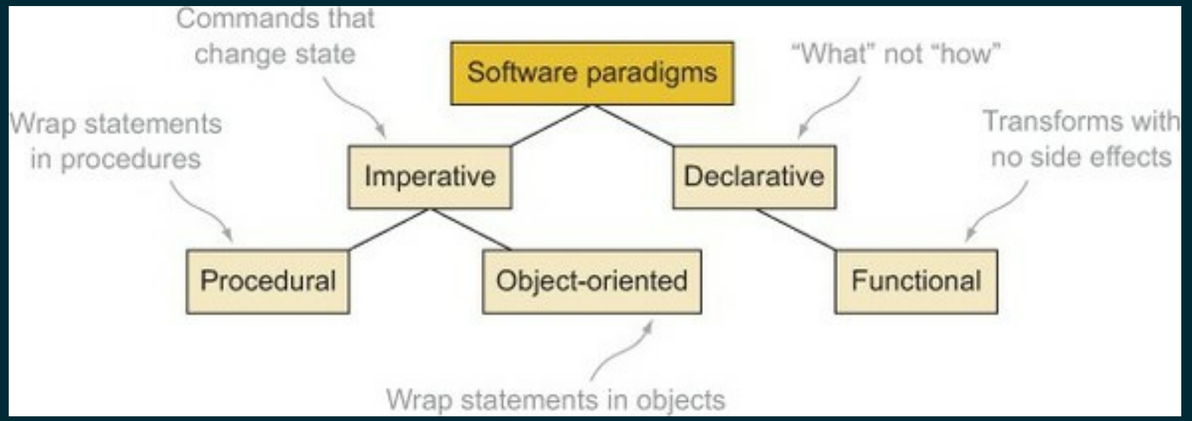
ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...

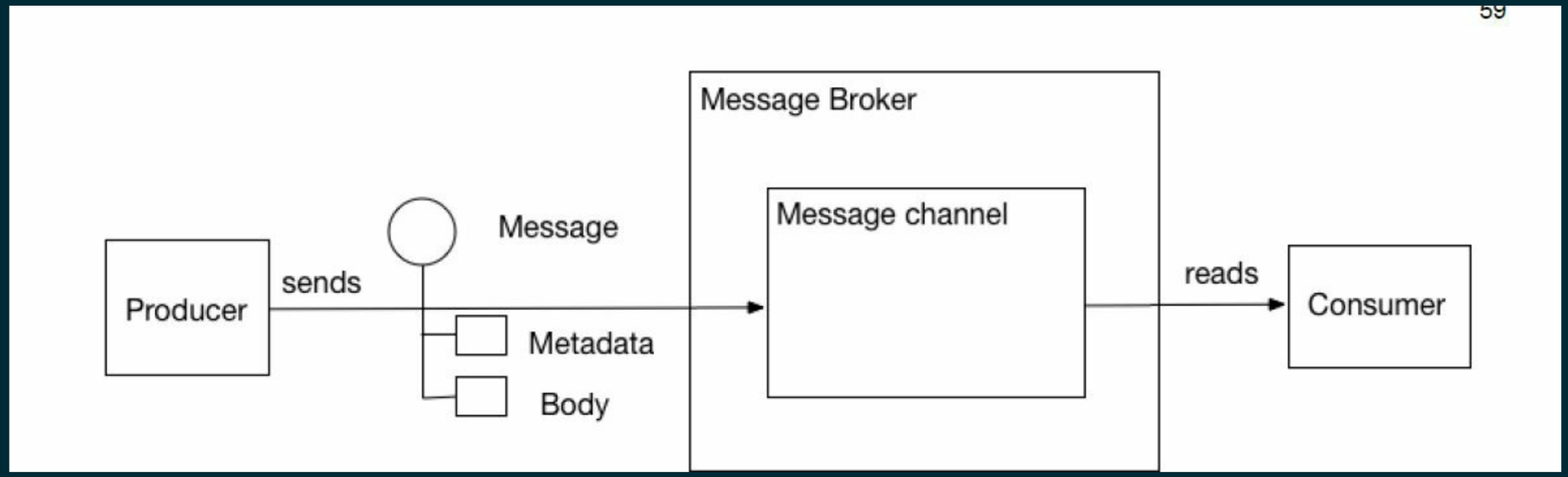


ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...

ARCHITECTURE & DESIGN ...



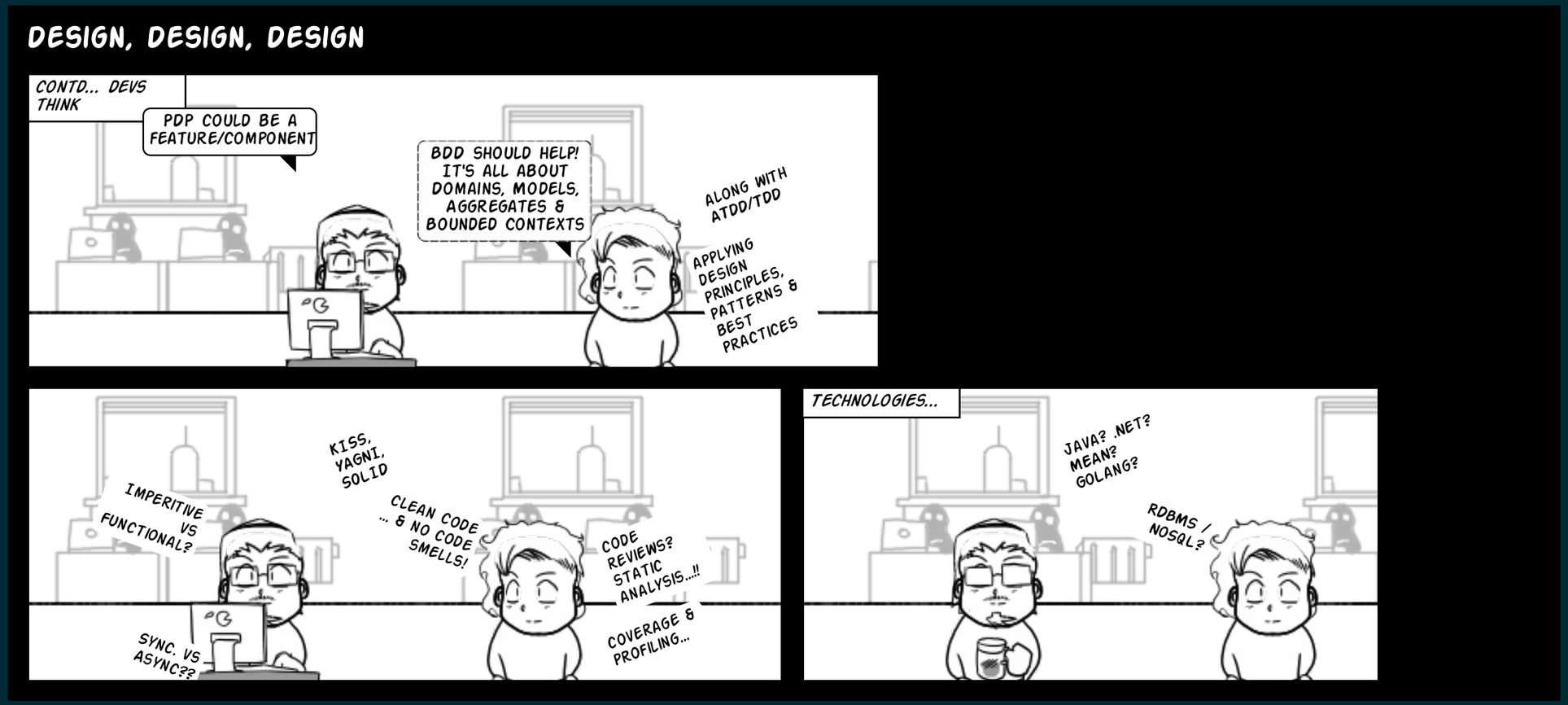
Synchronous does not perform all the time

AMQP based Prtocool Product Rabit MQ

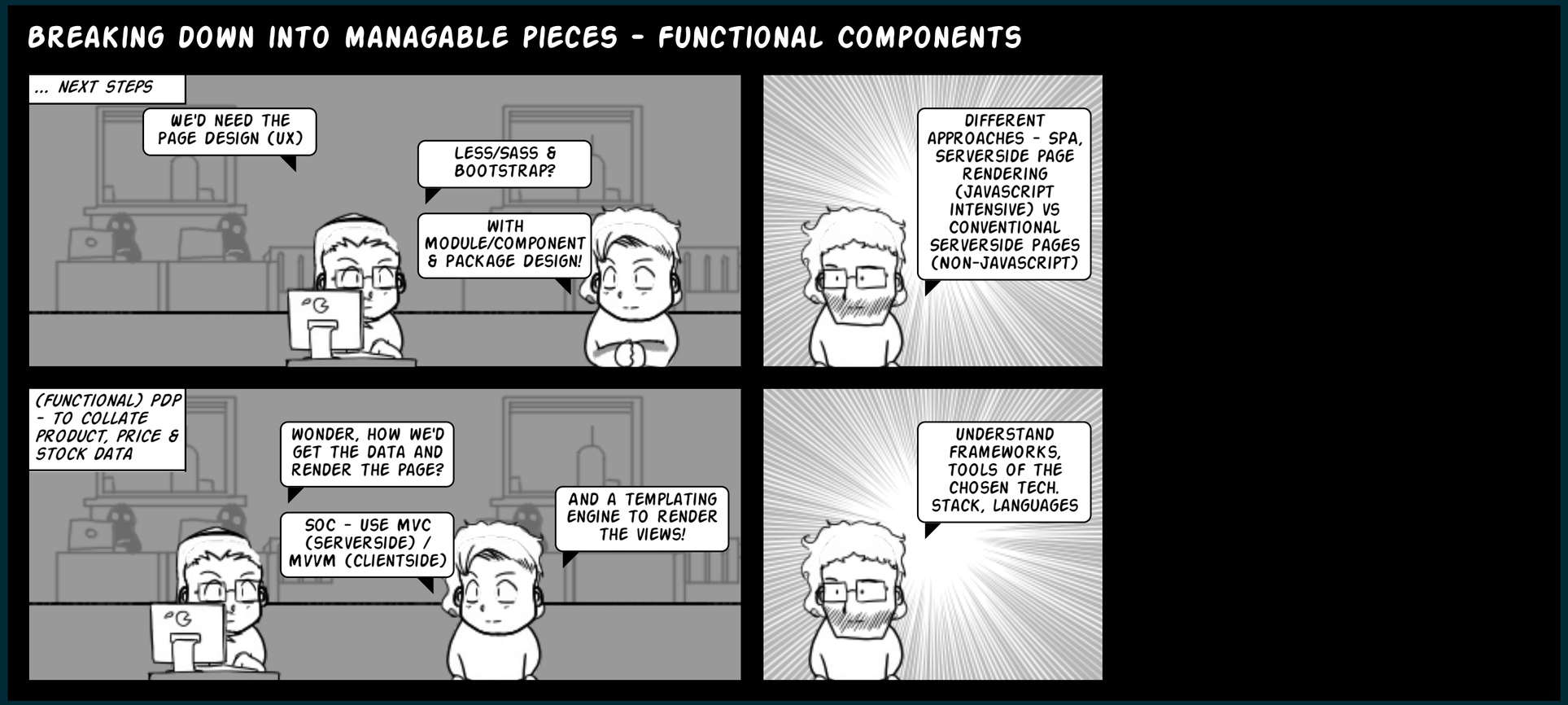
ARCHITECTURE & DESIGN ...

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

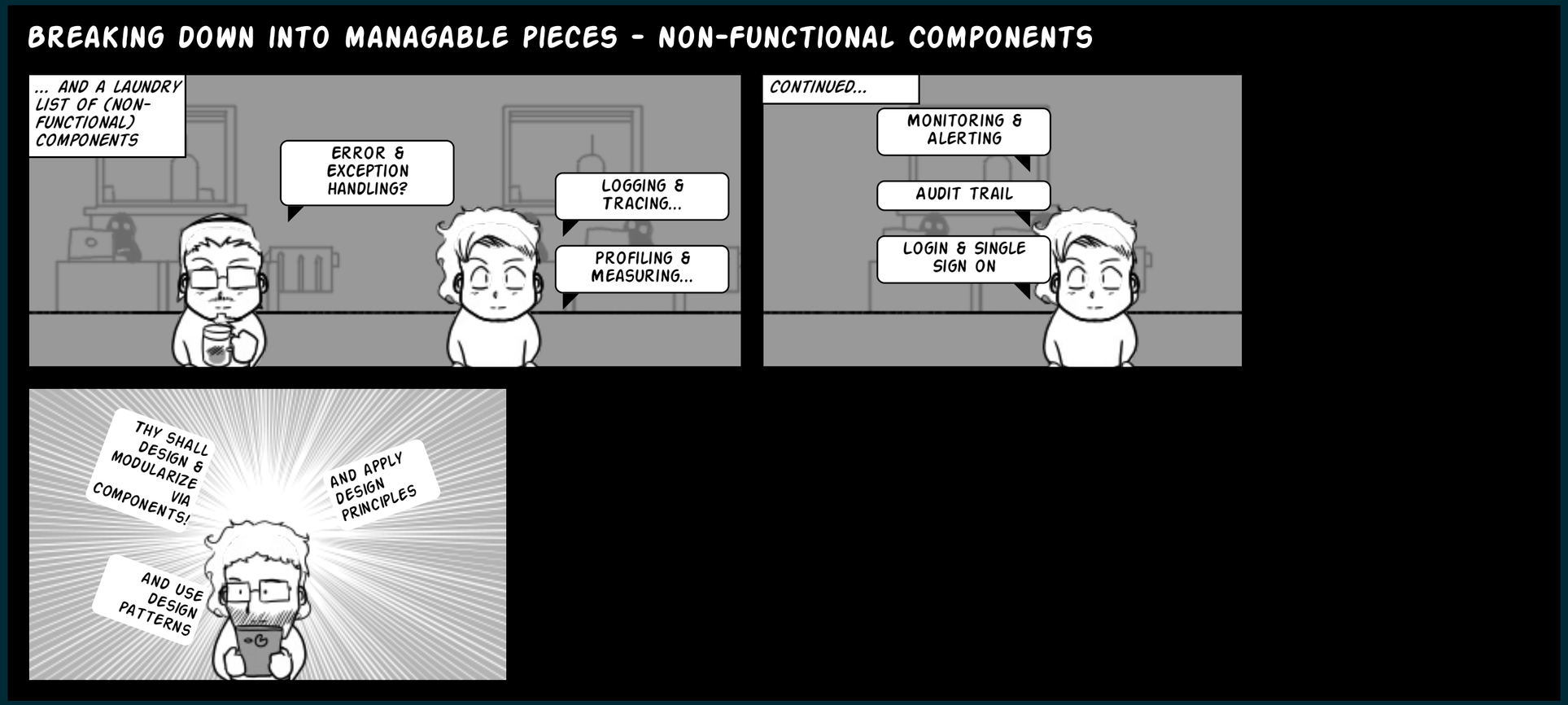
ARCHITECTURE & DESIGN ...



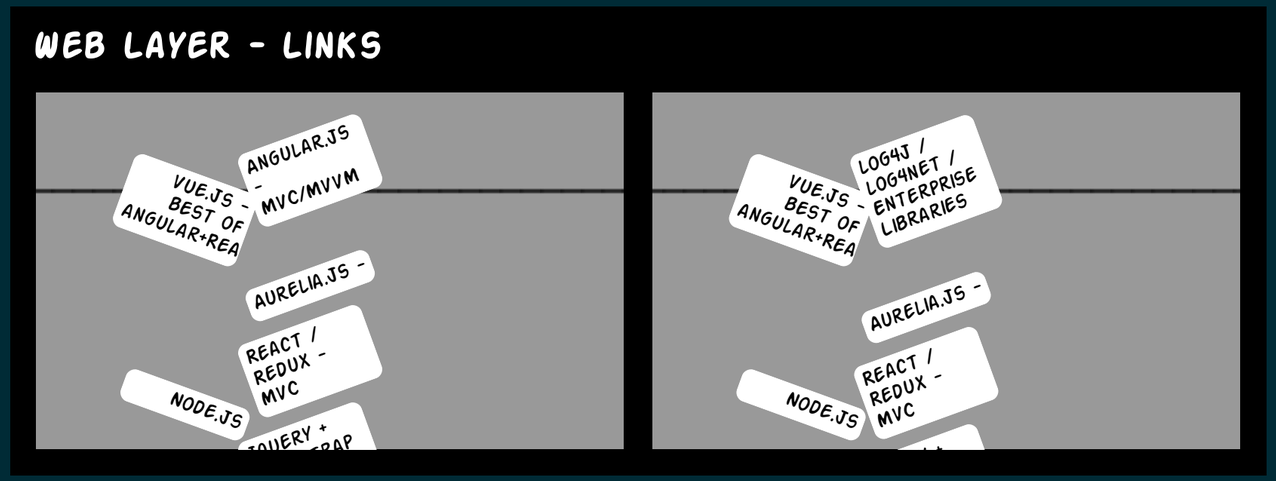
ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...

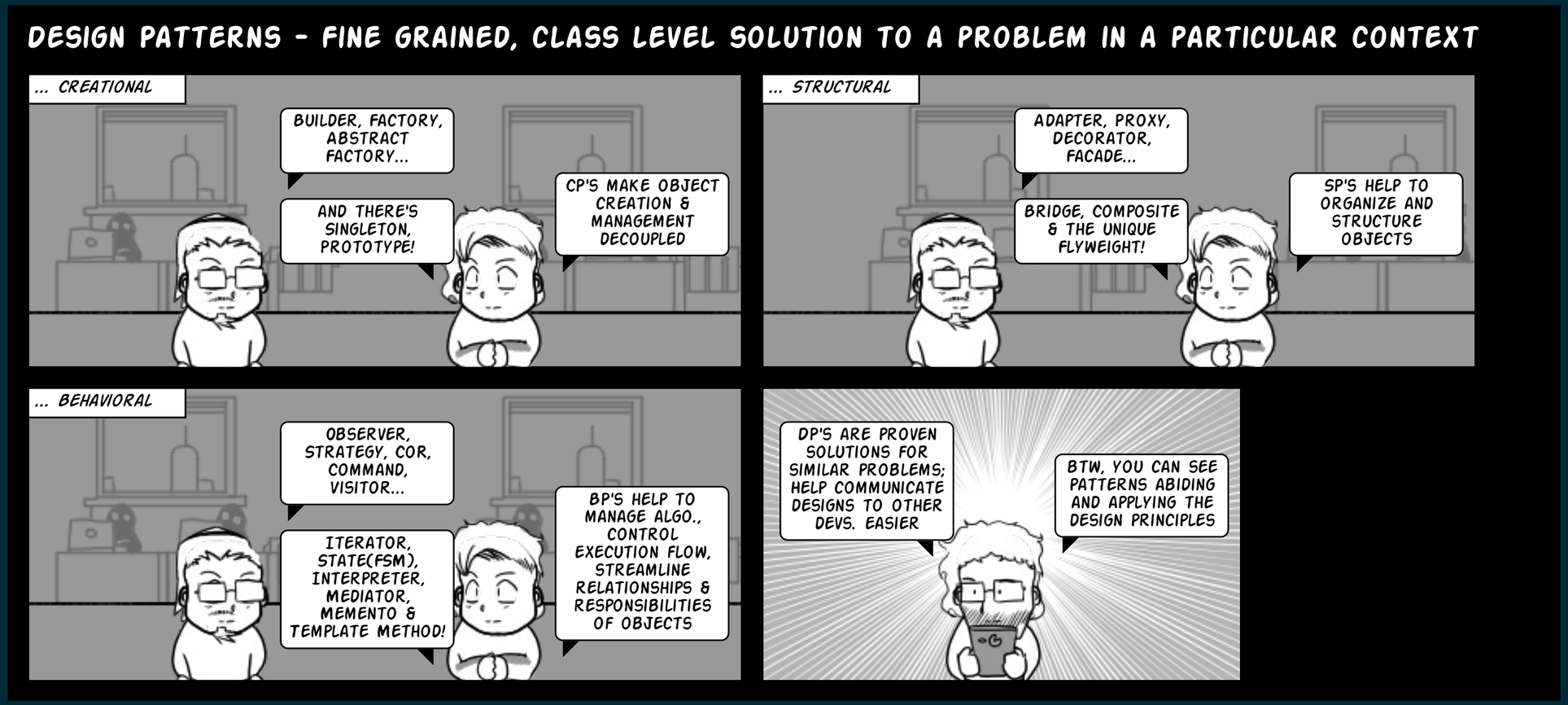


ARCHITECTURE & DESIGN ...

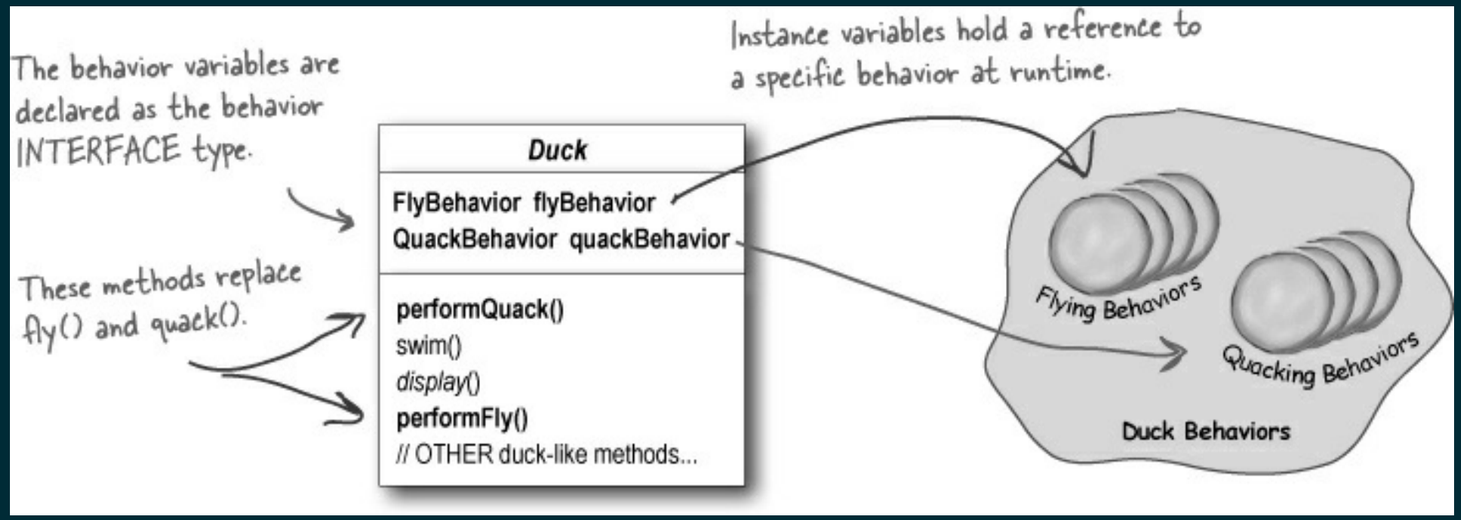
https://msdn.microsoft.com/en-us/library/ms998530.aspx
https://www.microsoft.com/en-us/download/confirmation.aspx?id=11711
https://msdn.microsoft.com/en-us/library/ff921345.aspx
https://msdn.microsoft.com/en-gb/library/ff648138.aspx
http://dataguidance.codeplex.com/releases

Architecture
http://shapingsoftware.com/
http://sourcesofinsight.com/

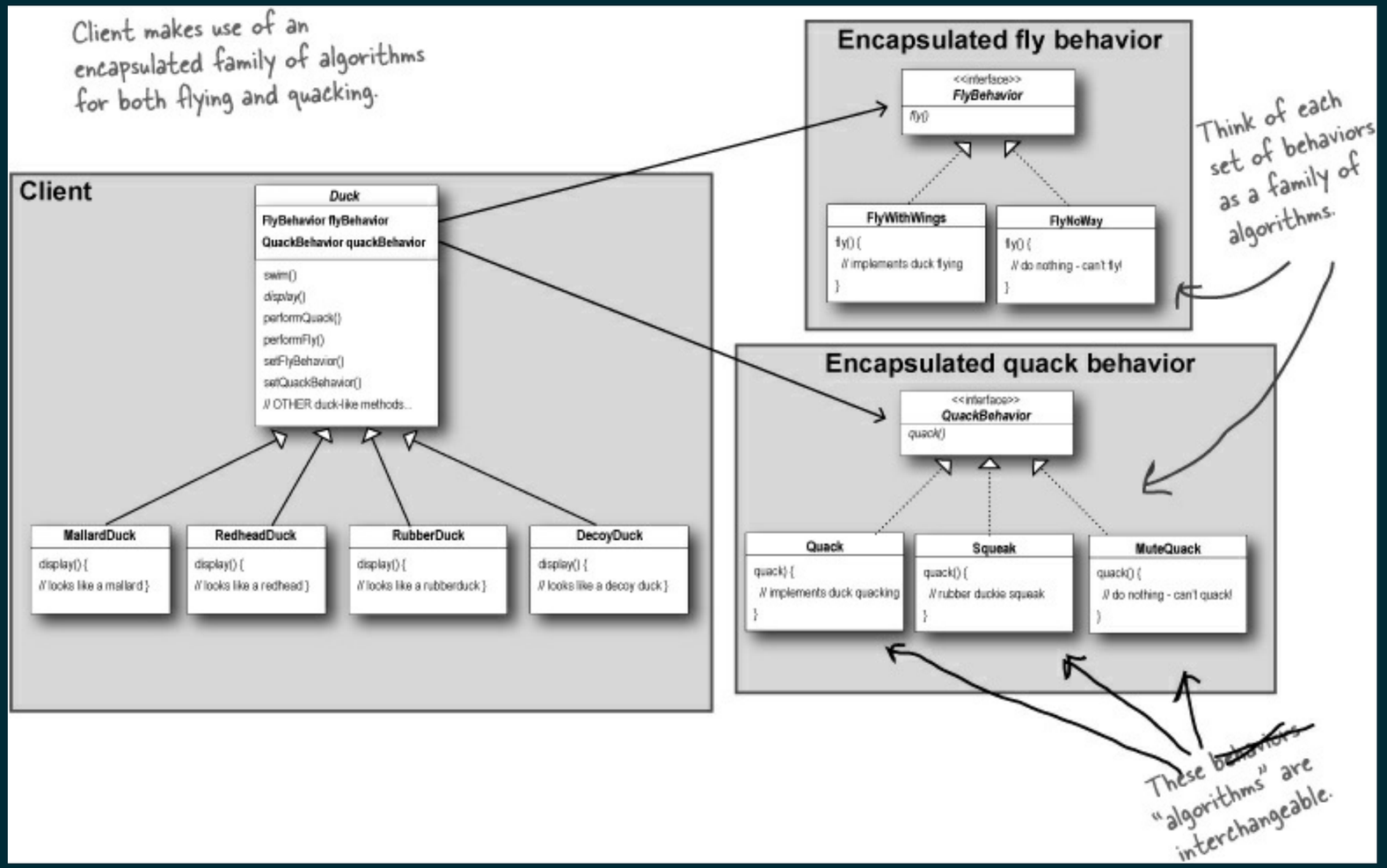
ARCHITECTURE & DESIGN ...



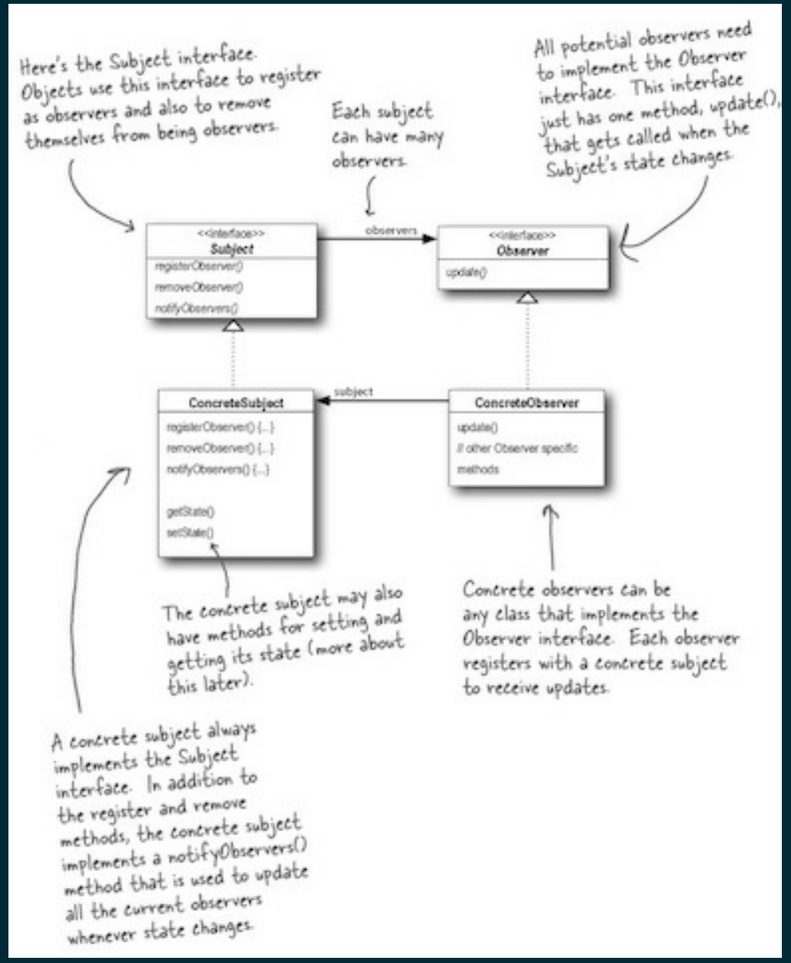
ARCHITECTURE & DESIGN ...



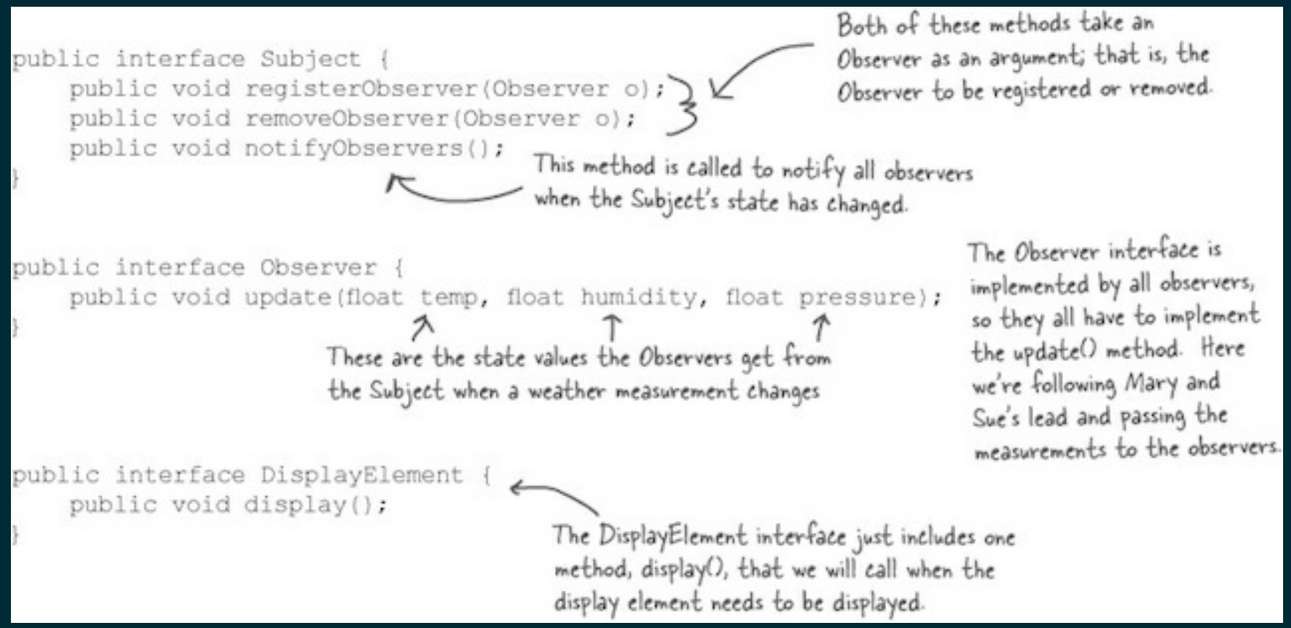
ARCHITECTURE & DESIGN ...



ARCHITECTURE & DESIGN ...



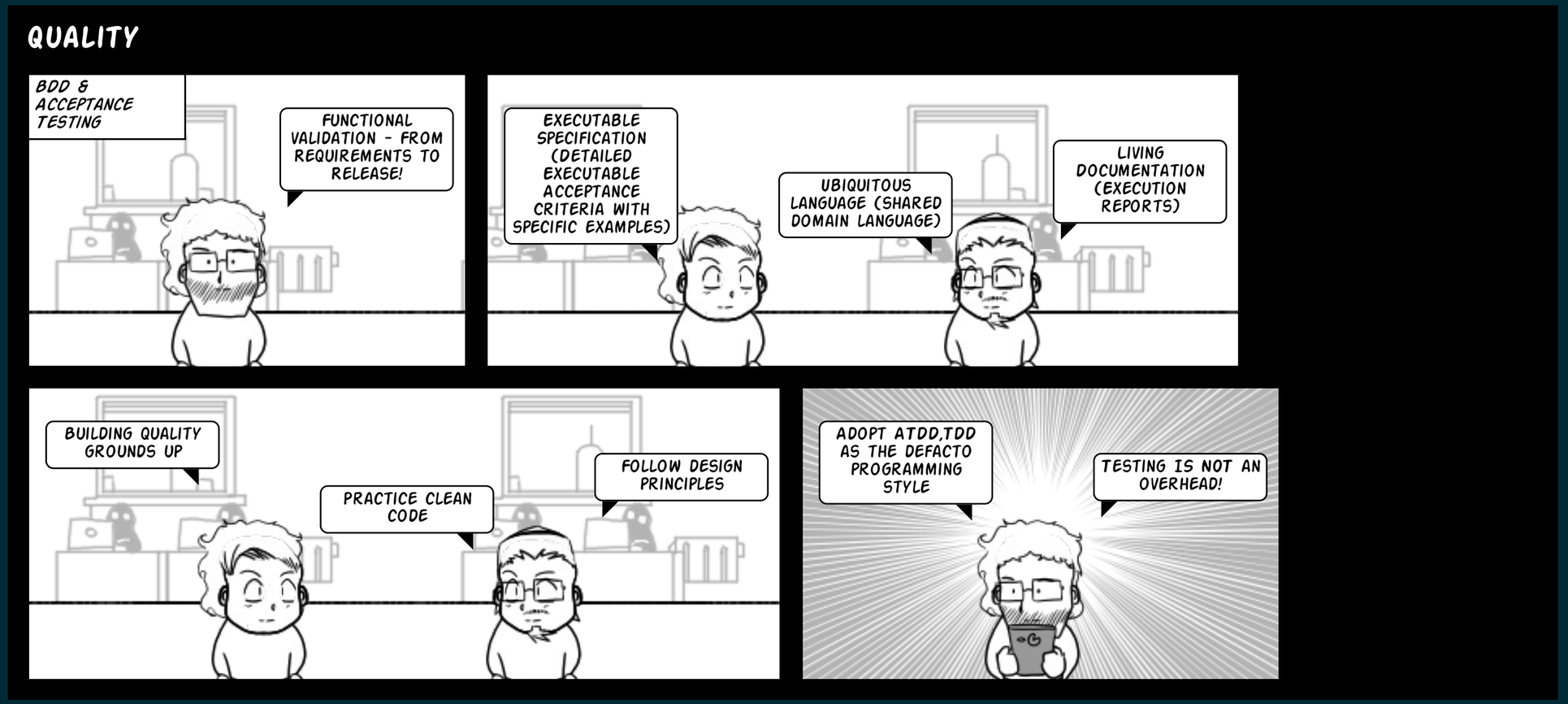
ARCHITECTURE & DESIGN ...



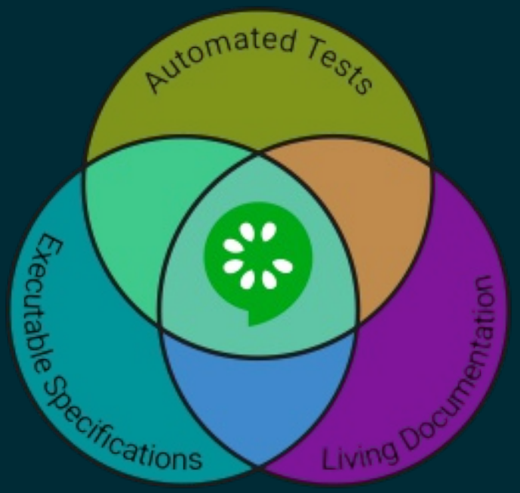
ARCHITECTURE & DESIGN ...

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

ENSURING QUALITY....



ENSURING QUALITY....



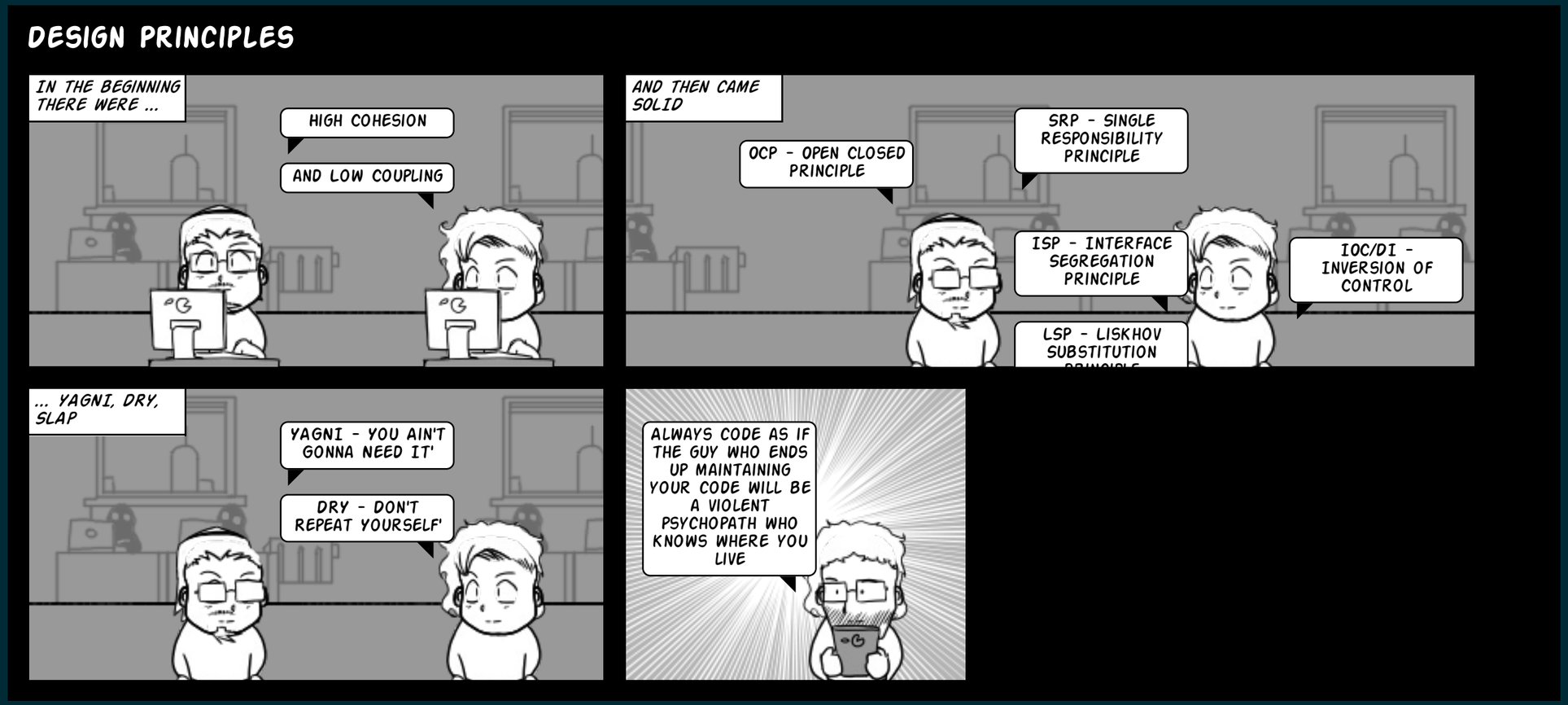
benefits of bdd

BDD Frameworks

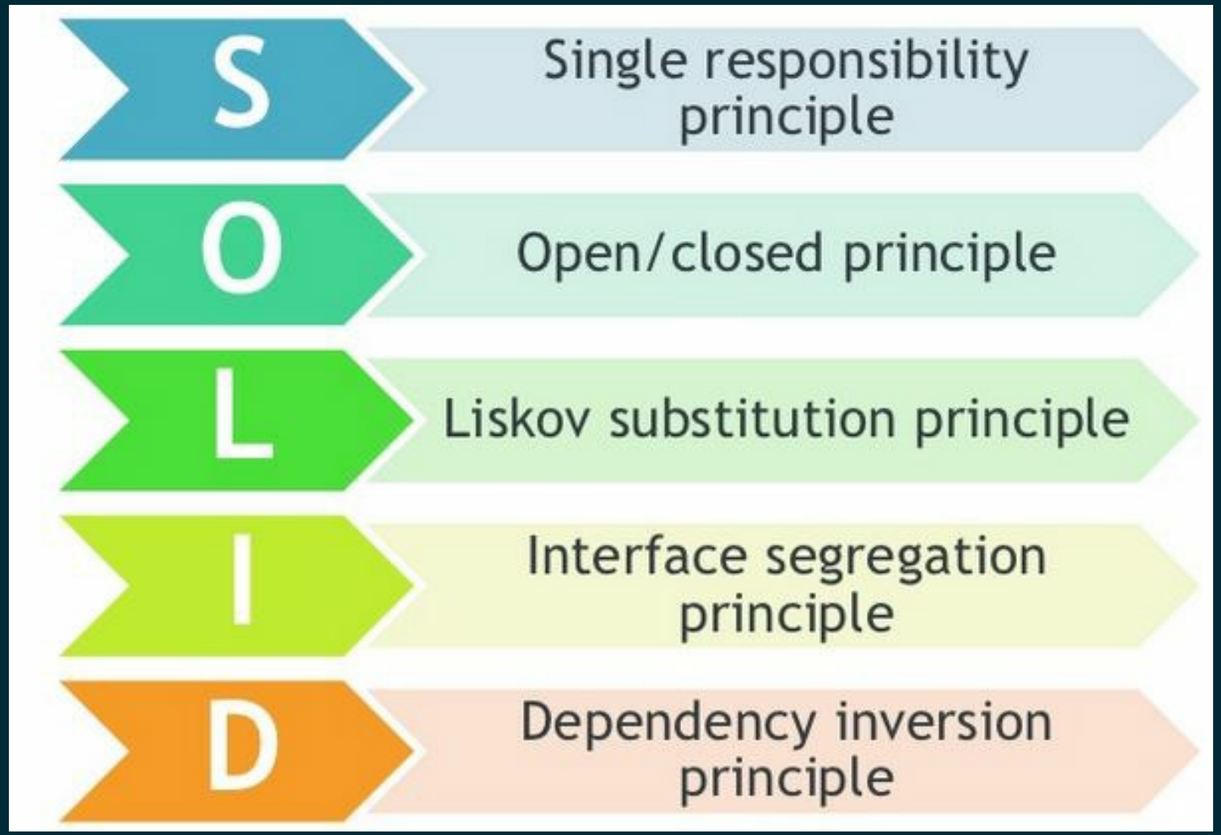
Book: BDD in Action

BDD - Introduction from Dan North

ENSURING QUALITY....



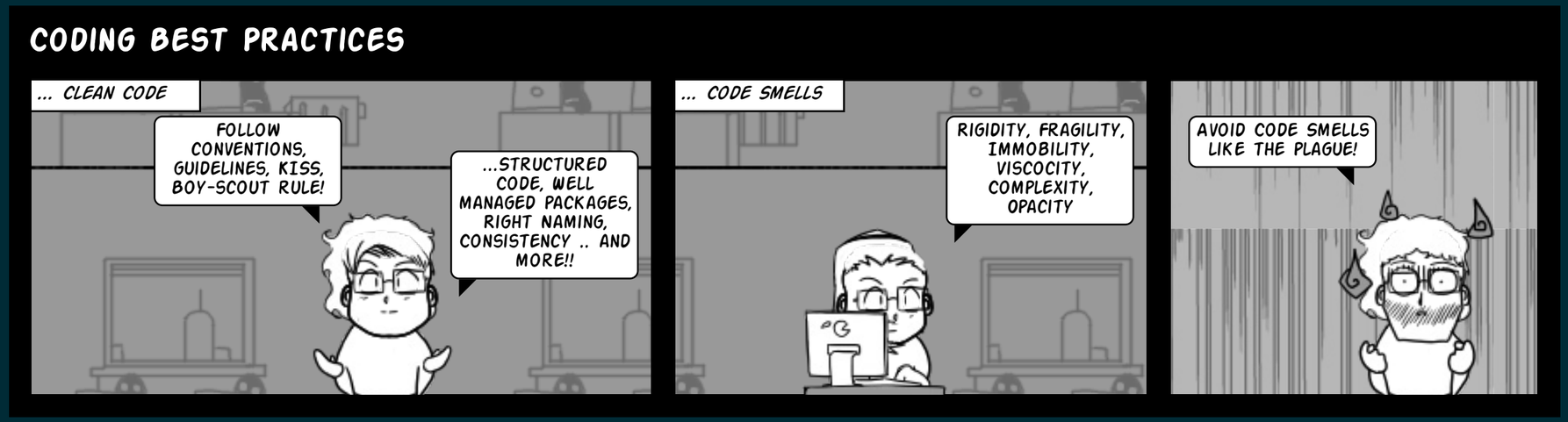
ENSURING QUALITY....



ENSURING QUALITY....

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

ENSURING QUALITY....

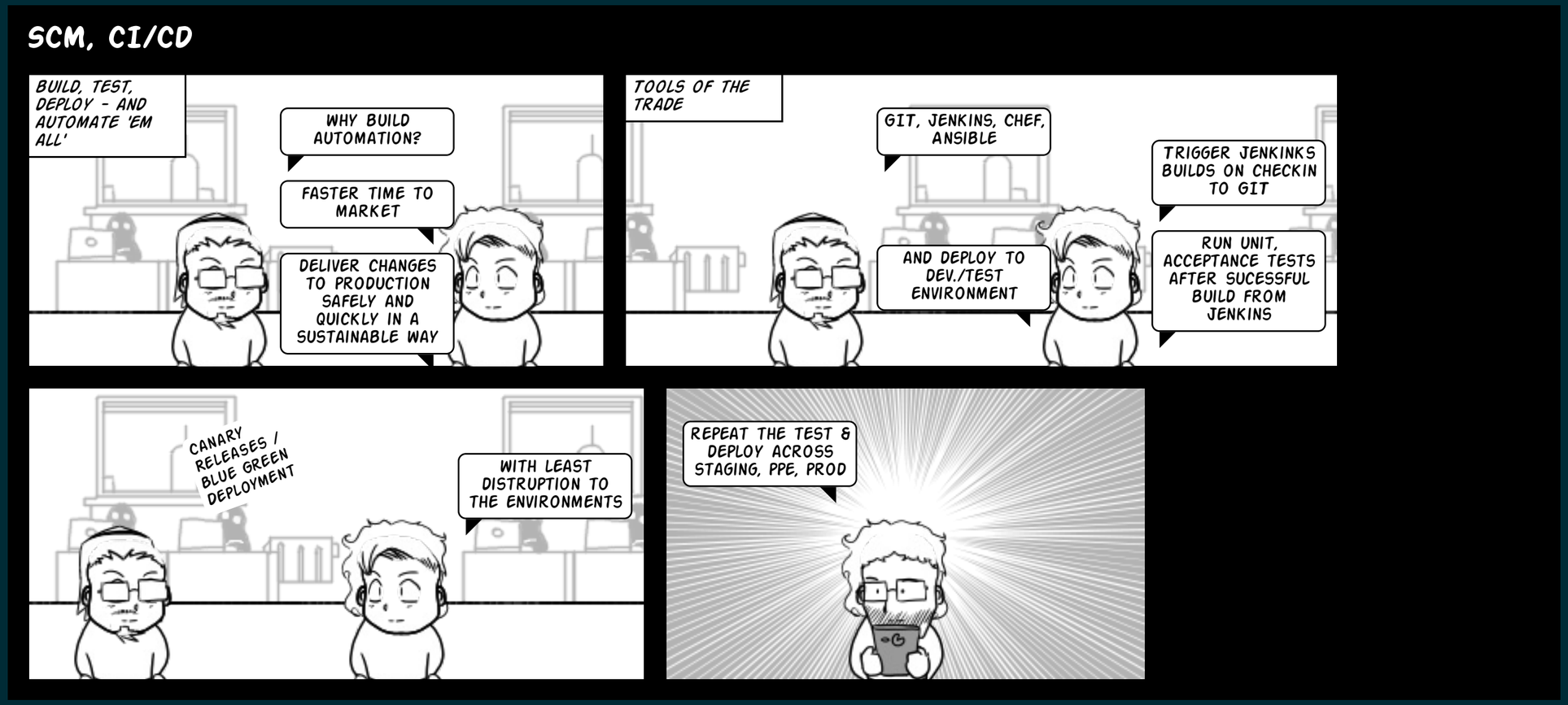


ENSURING QUALITY....

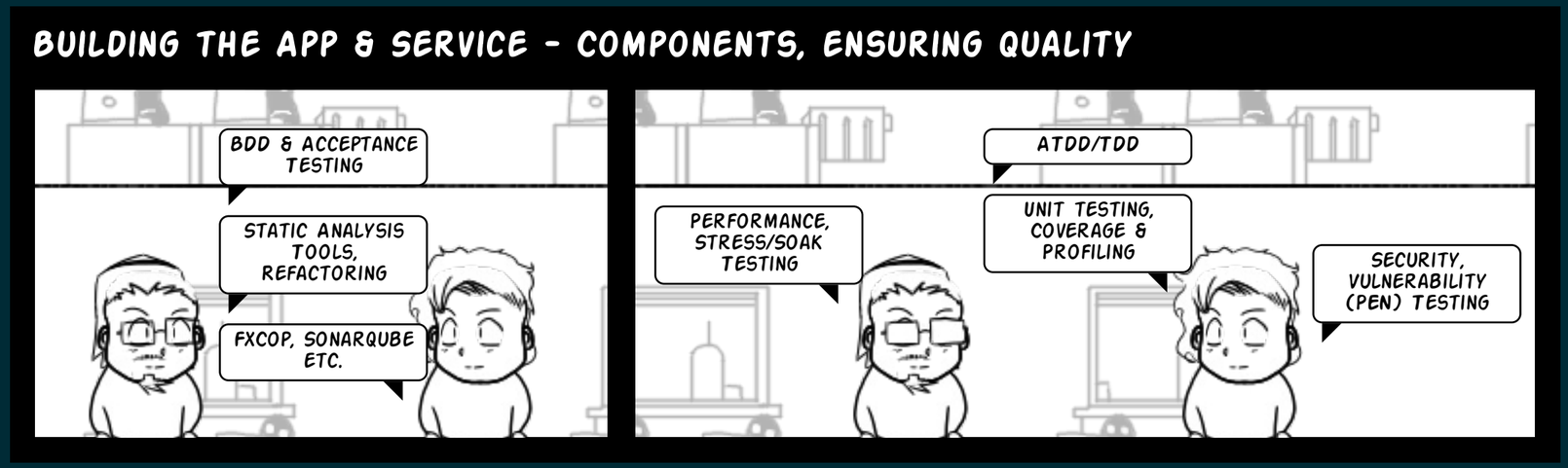
http://www.planetgeek.ch/wp-content/uploads/2013/06/Clean-Code-V2.1.pdf
https://www.amazon.in/Clean-Code-Handbook-Software-Craftsmanship-ebook/dp/B001GSTOAM
http://www.hanselman.com/blog/SixEssentialLanguageAgnosticProgrammingBooks.aspx
https://github.com/chhantyal/influential-cs-books

ENSURING QUALITY....

ENSURING QUALITY....

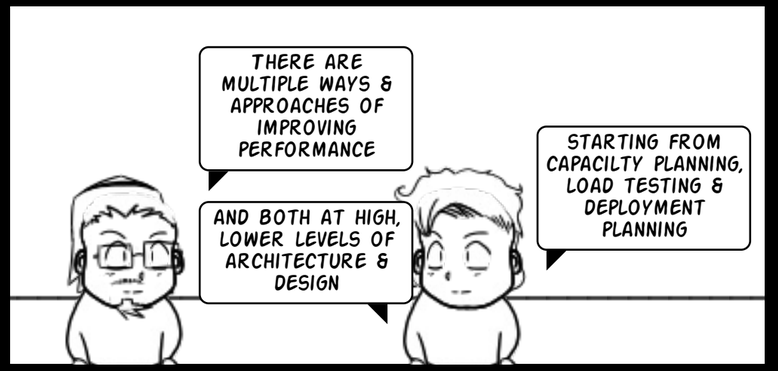


ENSURING QUALITY....



PERFORMANCE

WHY IMPROVE PERFORMANCE?

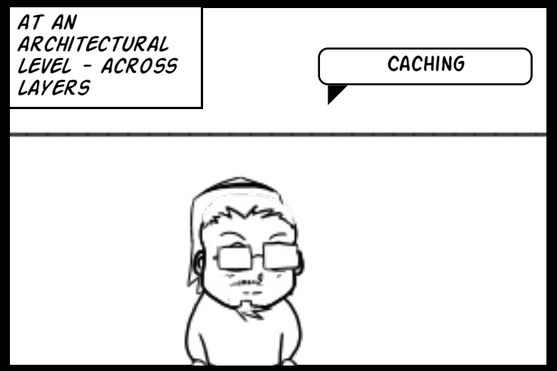


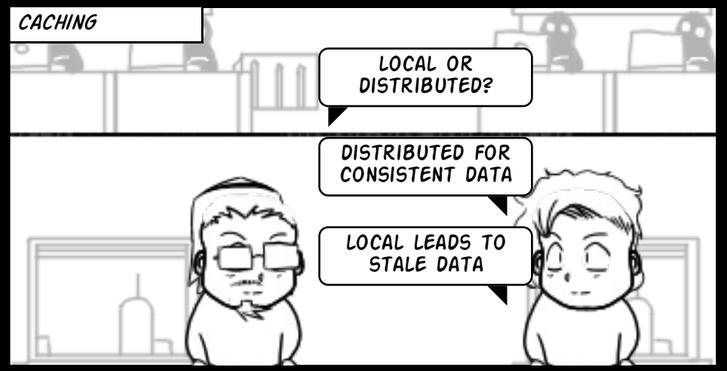
PERFORMANCE

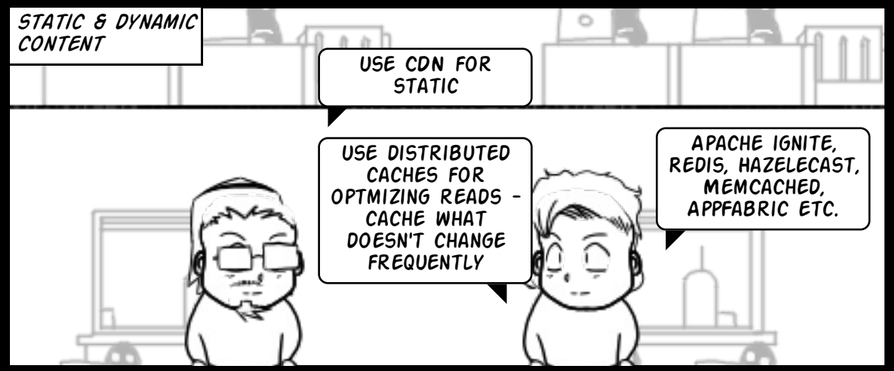
http://perftesting.codeplex.com/
http://guidanceengineering.codeplex.com/
https://msdn.microsoft.com/en-us/library/ms998408.aspx
https://msdn.microsoft.com/practices

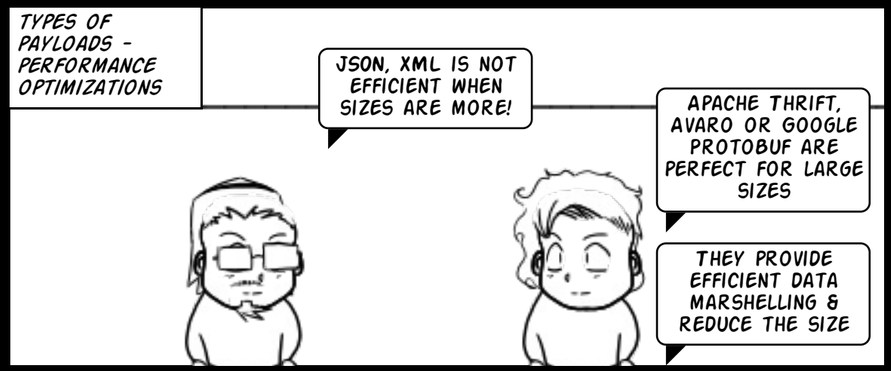
PERFORMANCE

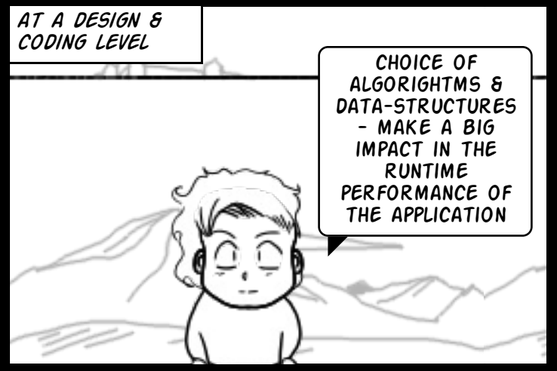
HOW TO IMPROVE PERFORMANCE



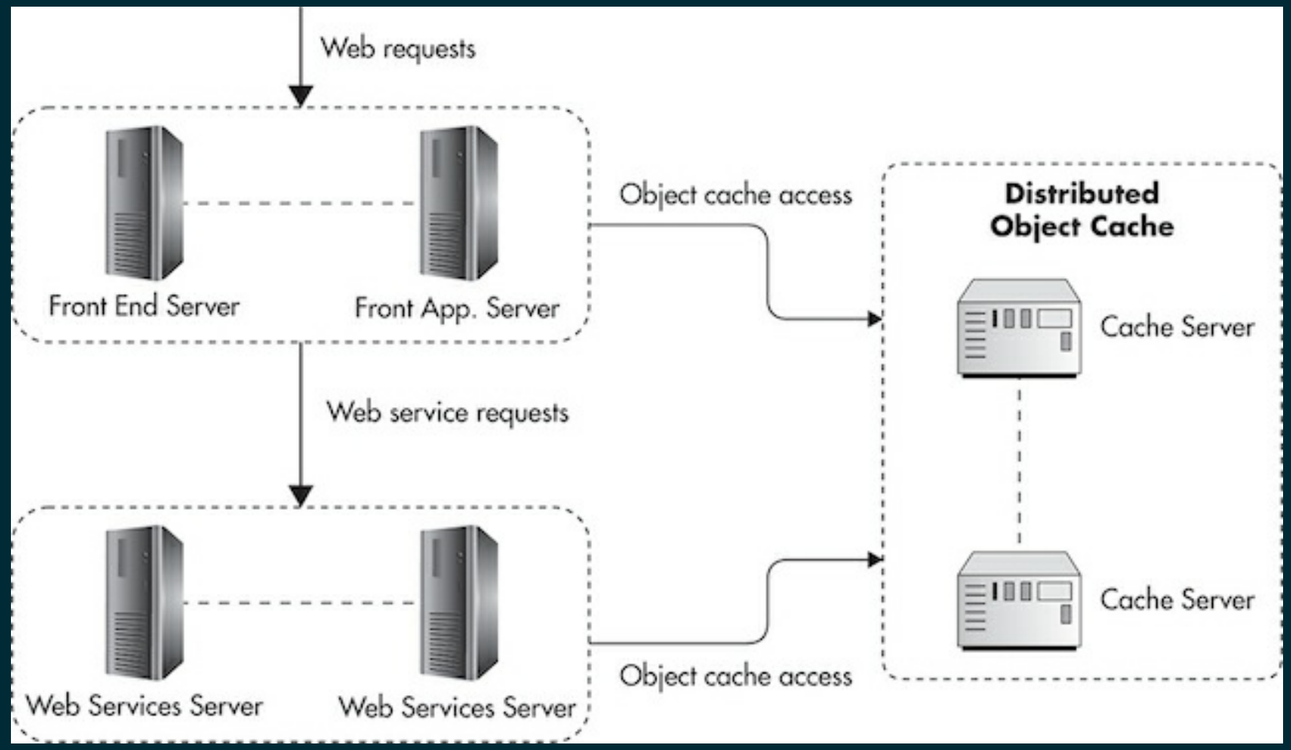






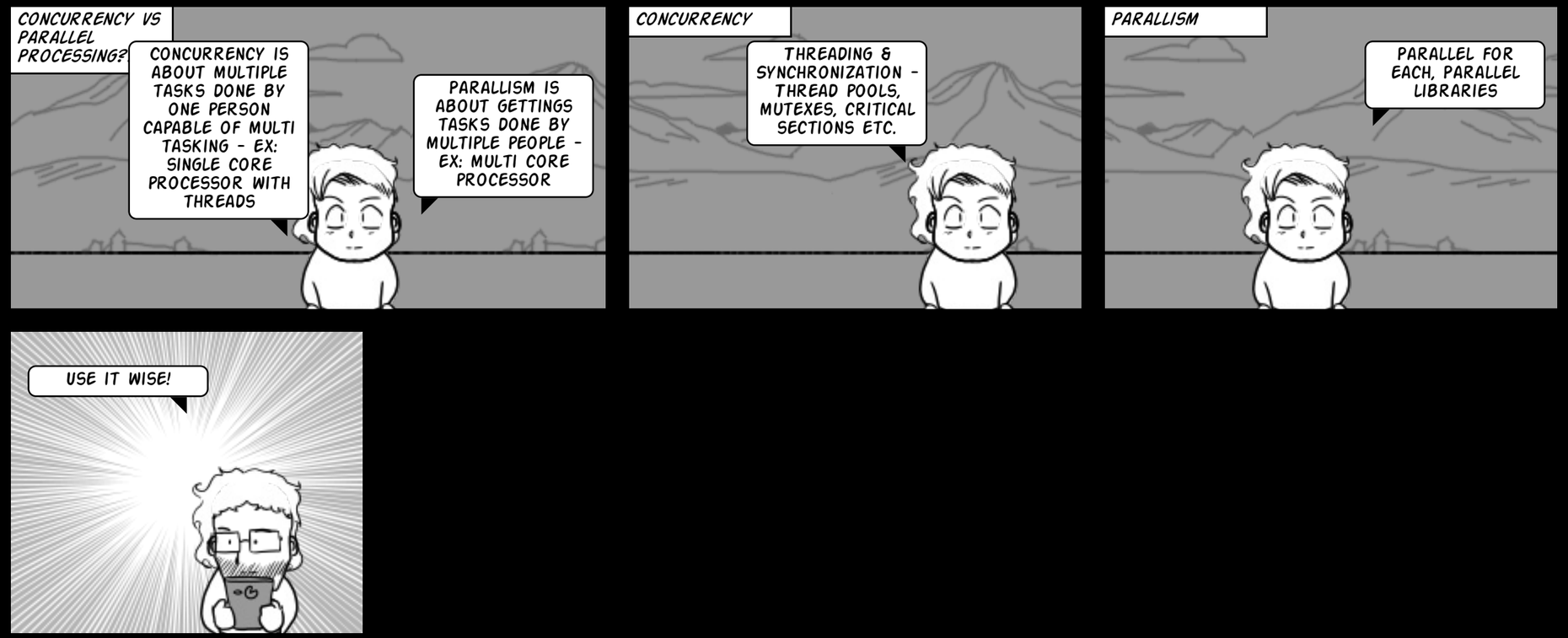


PERFORMANCE

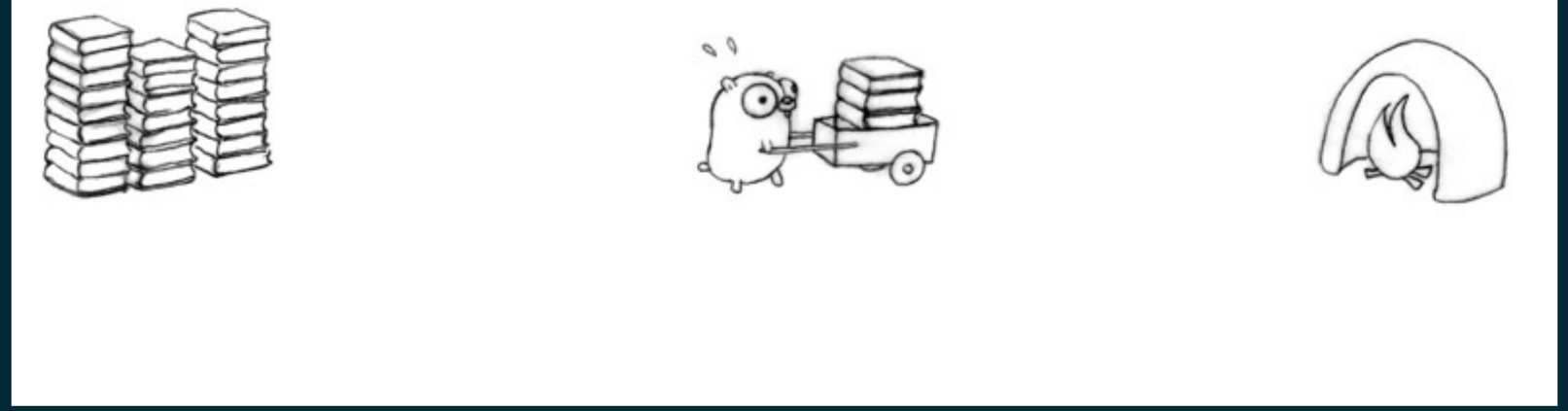


PERFORMANCE

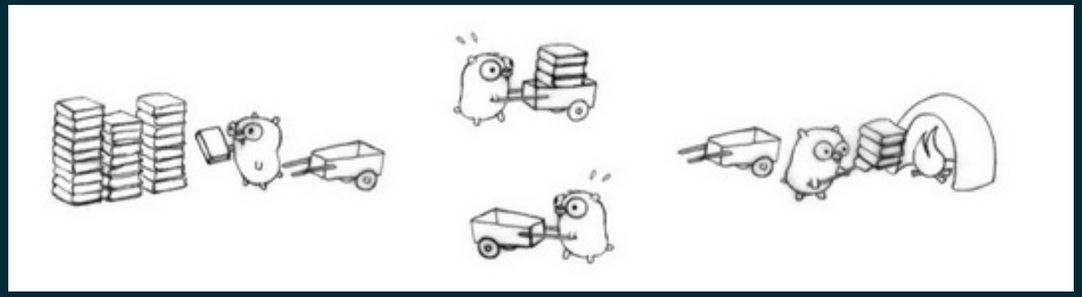
CONCURRENCY & PARALLELISM



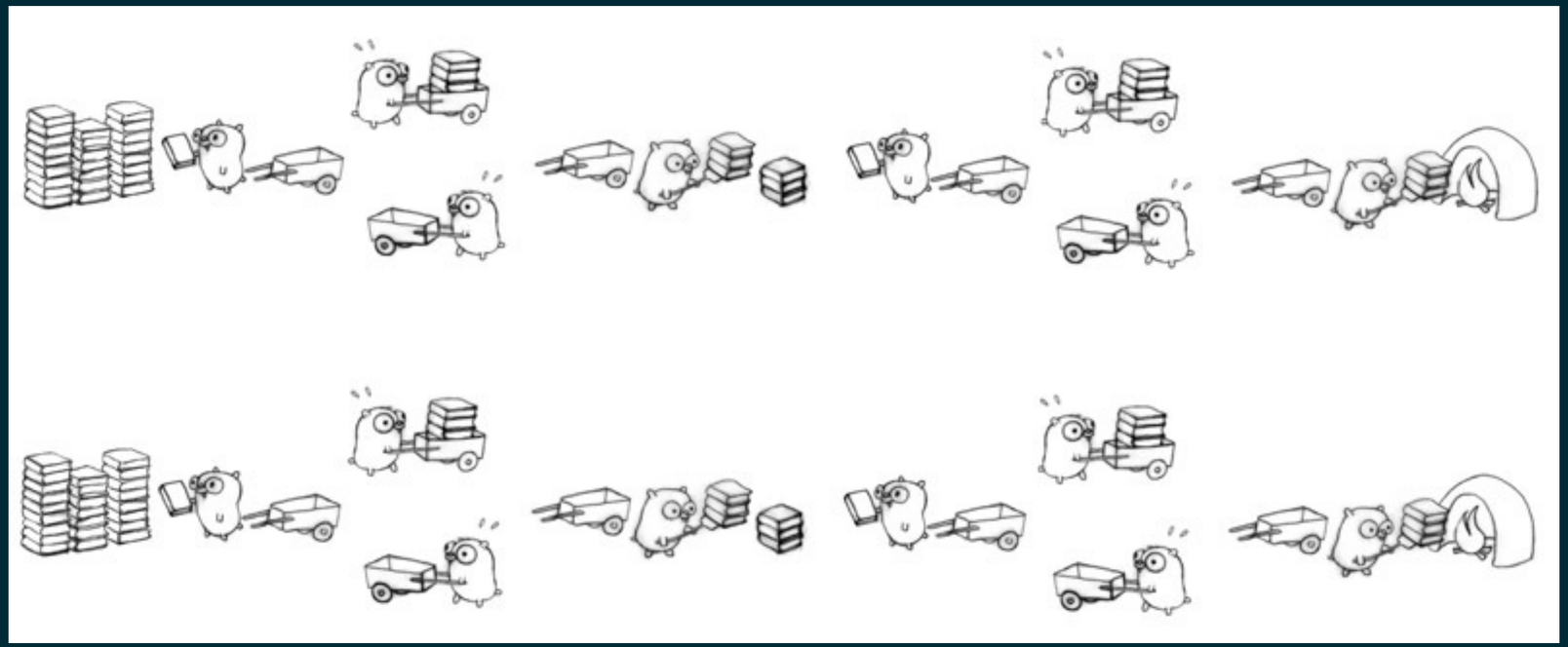
PERFORMANCE



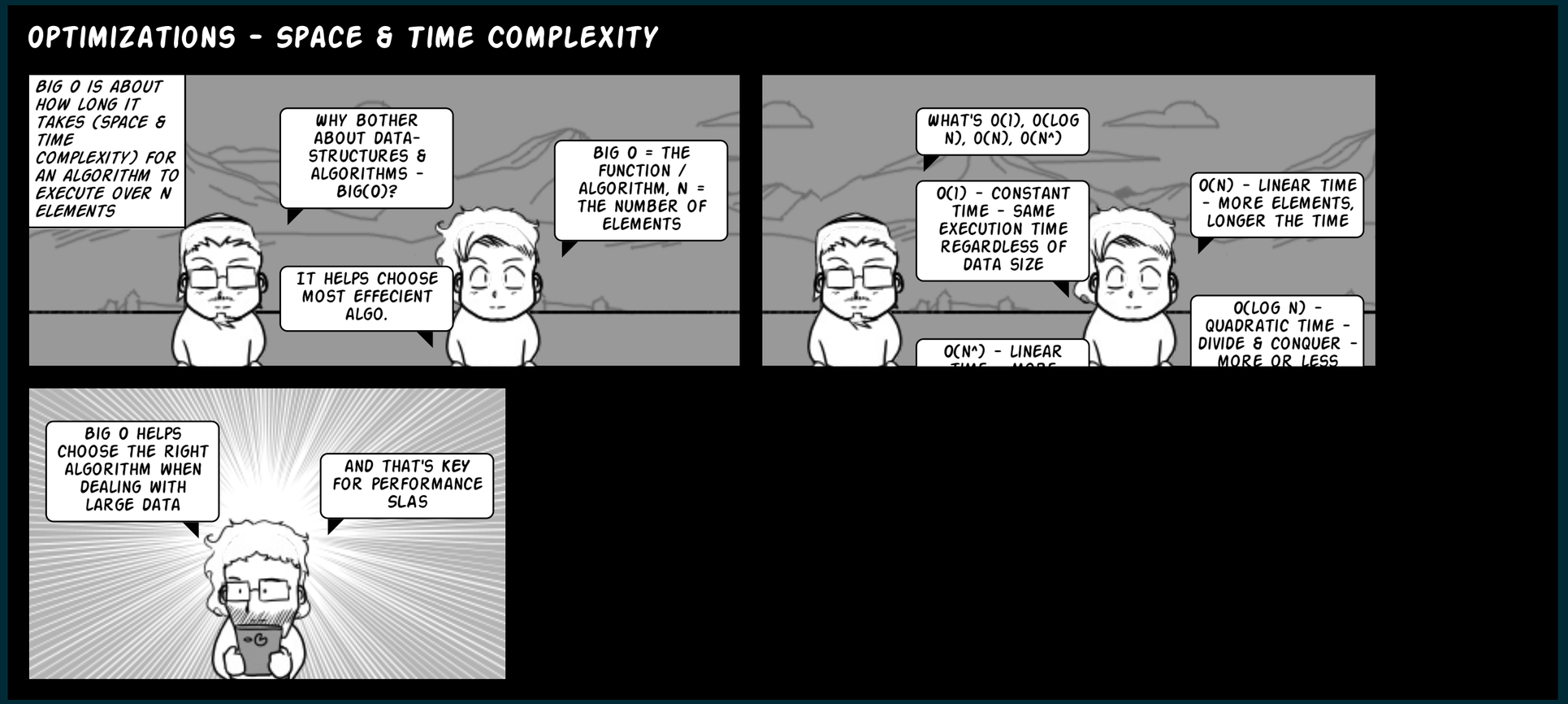
PERFORMANCE



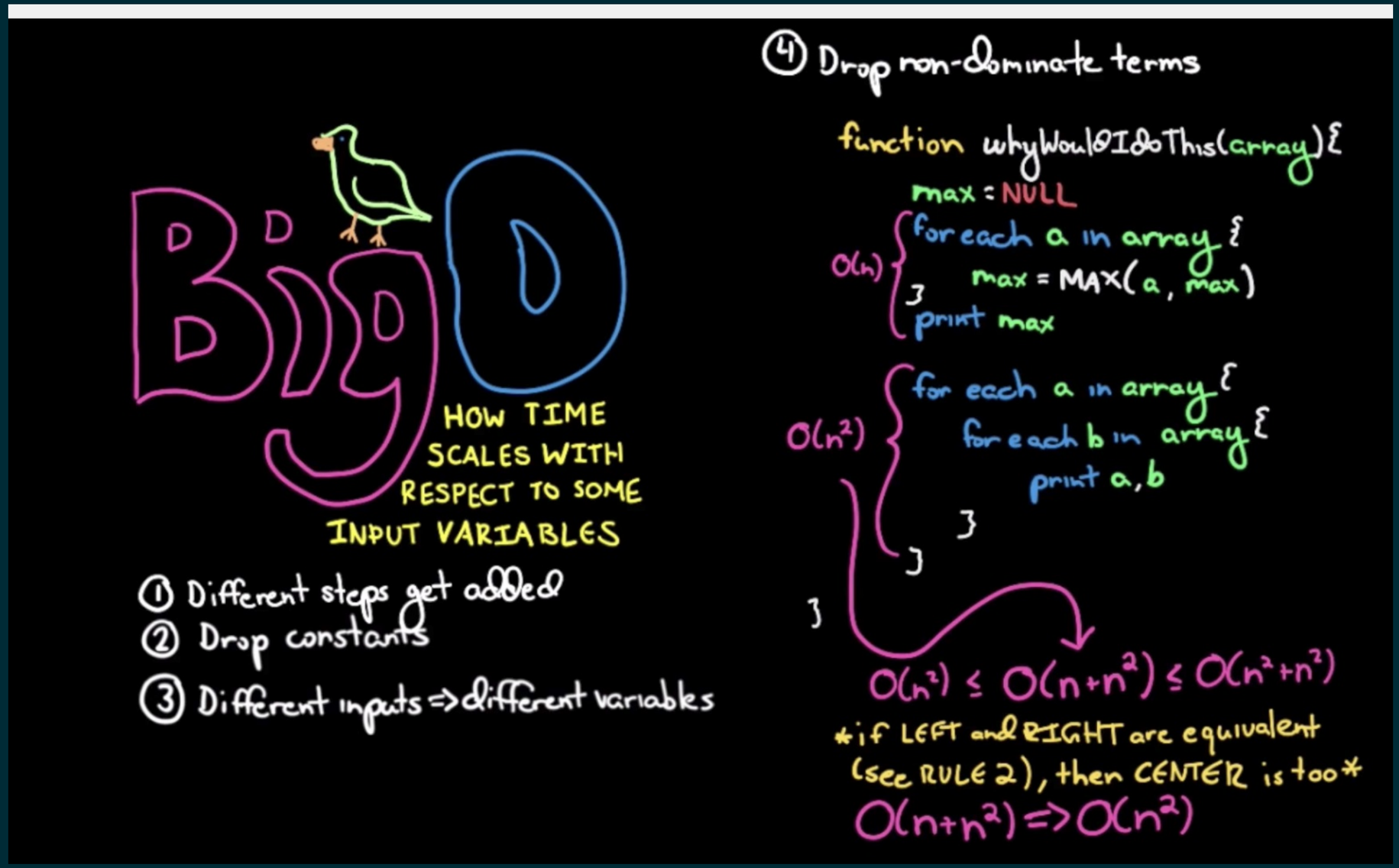
PERFORMANCE



PERFORMANCE



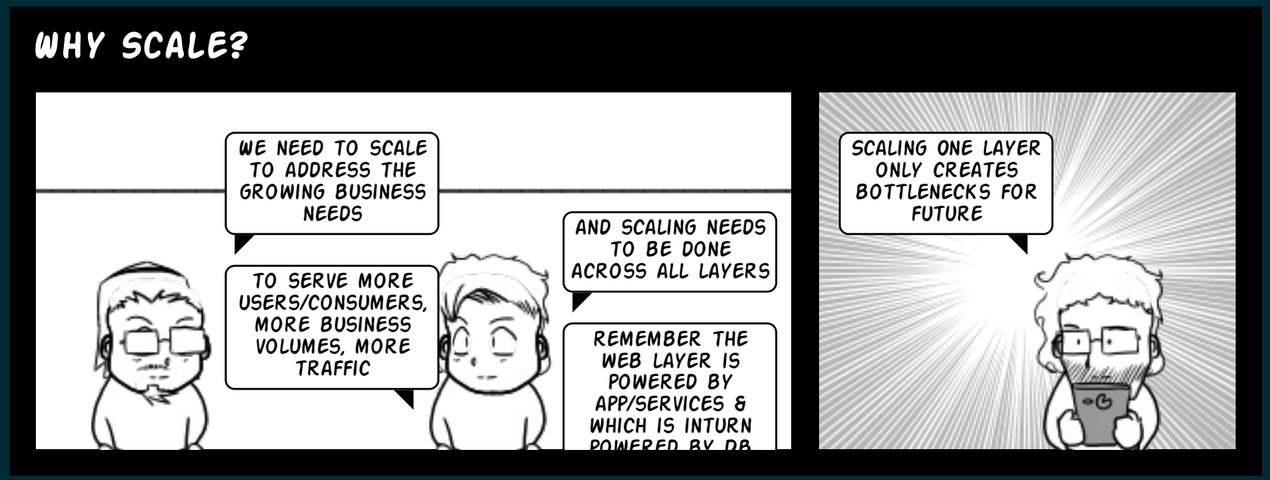
PERFORMANCE



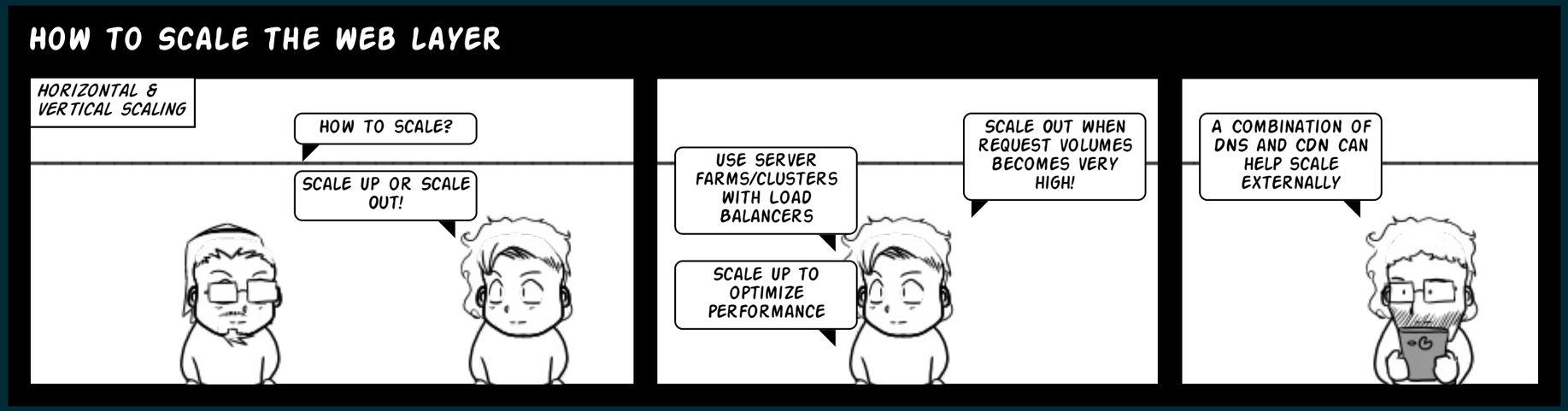
PERFORMANCE

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-sortin
g/
http://algorithmiccomplexity.com/
http://web.mit.edu/16.070/www/lecture/big_o.pdf

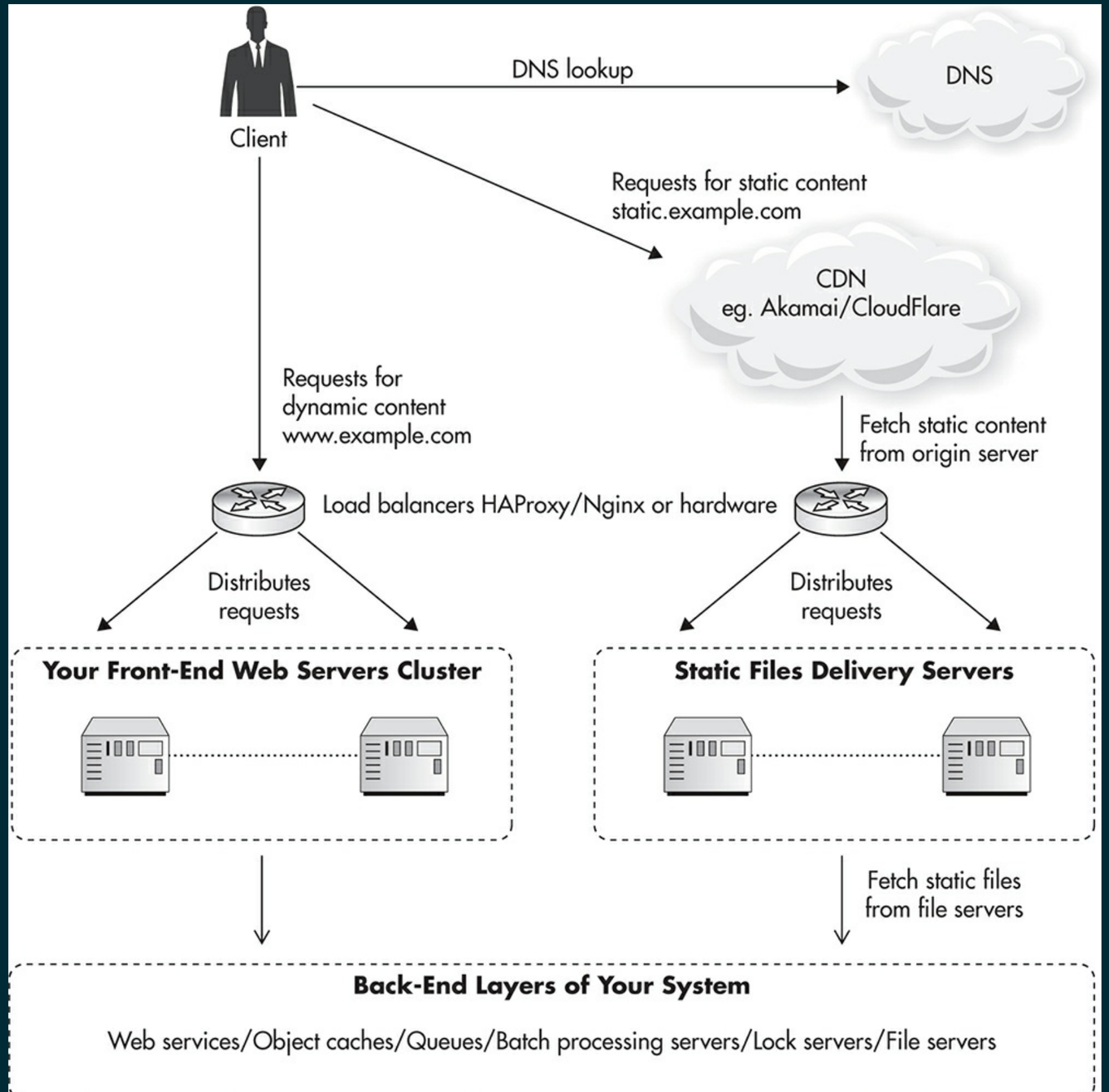
SCALABILITY



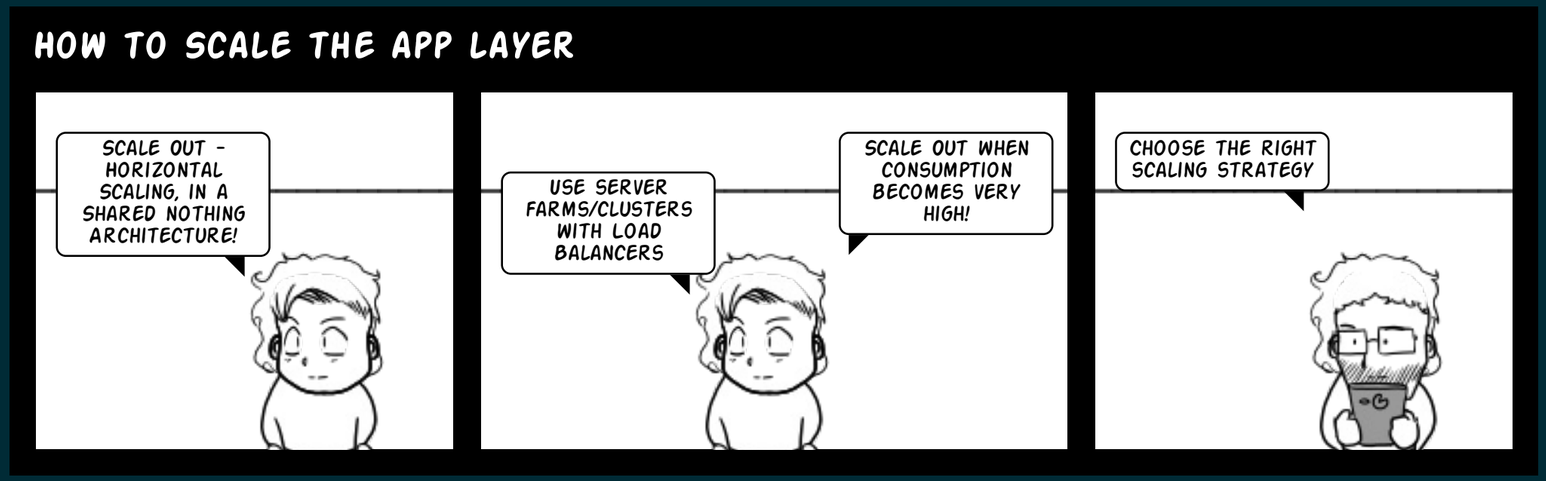
SCALABILITY



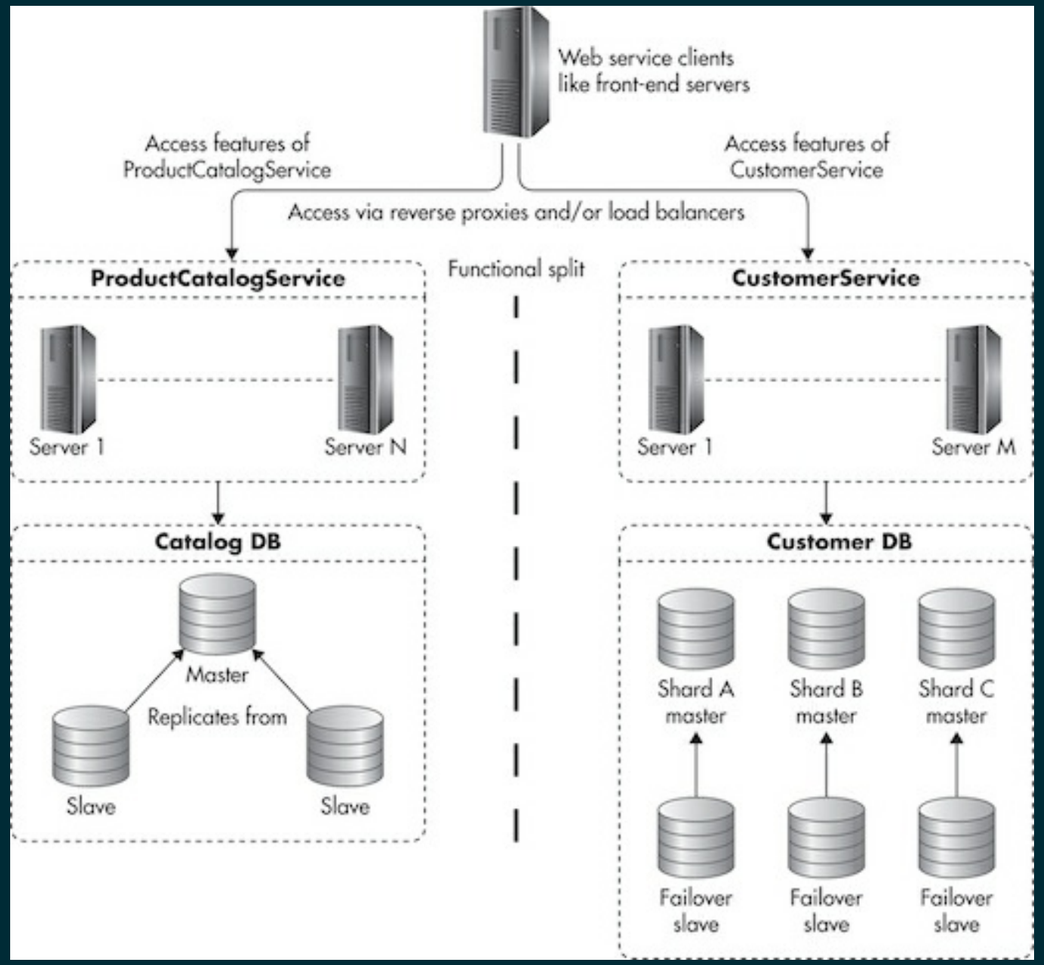
SCALABILITY



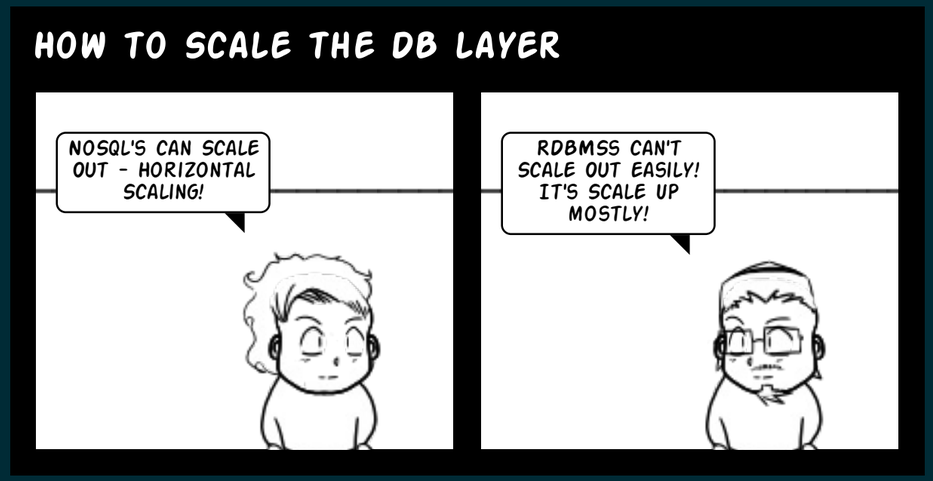
SCALABILITY



SCALABILITY

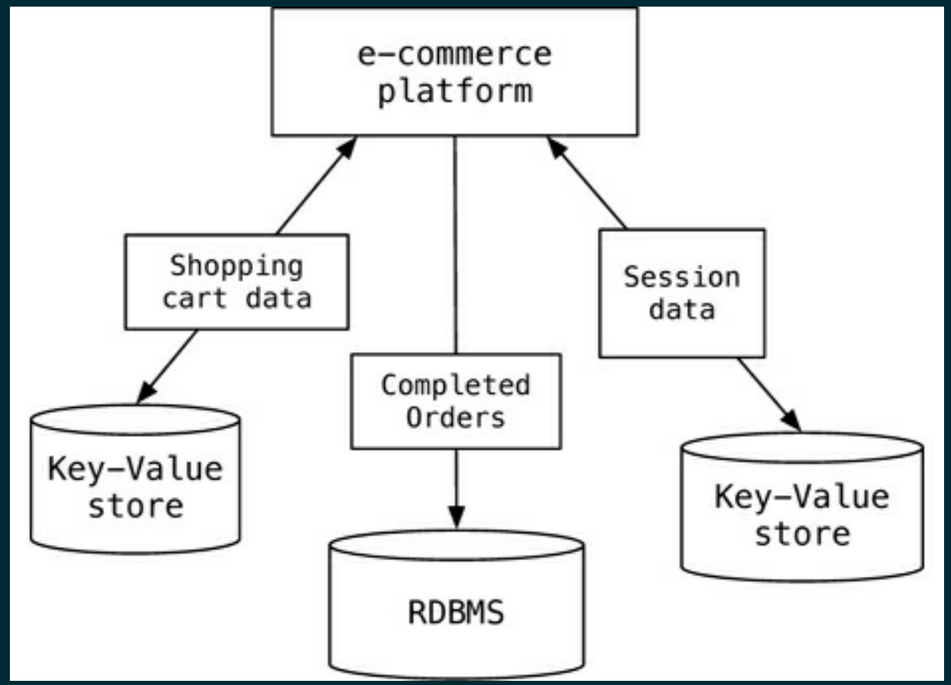


SCALABILITY



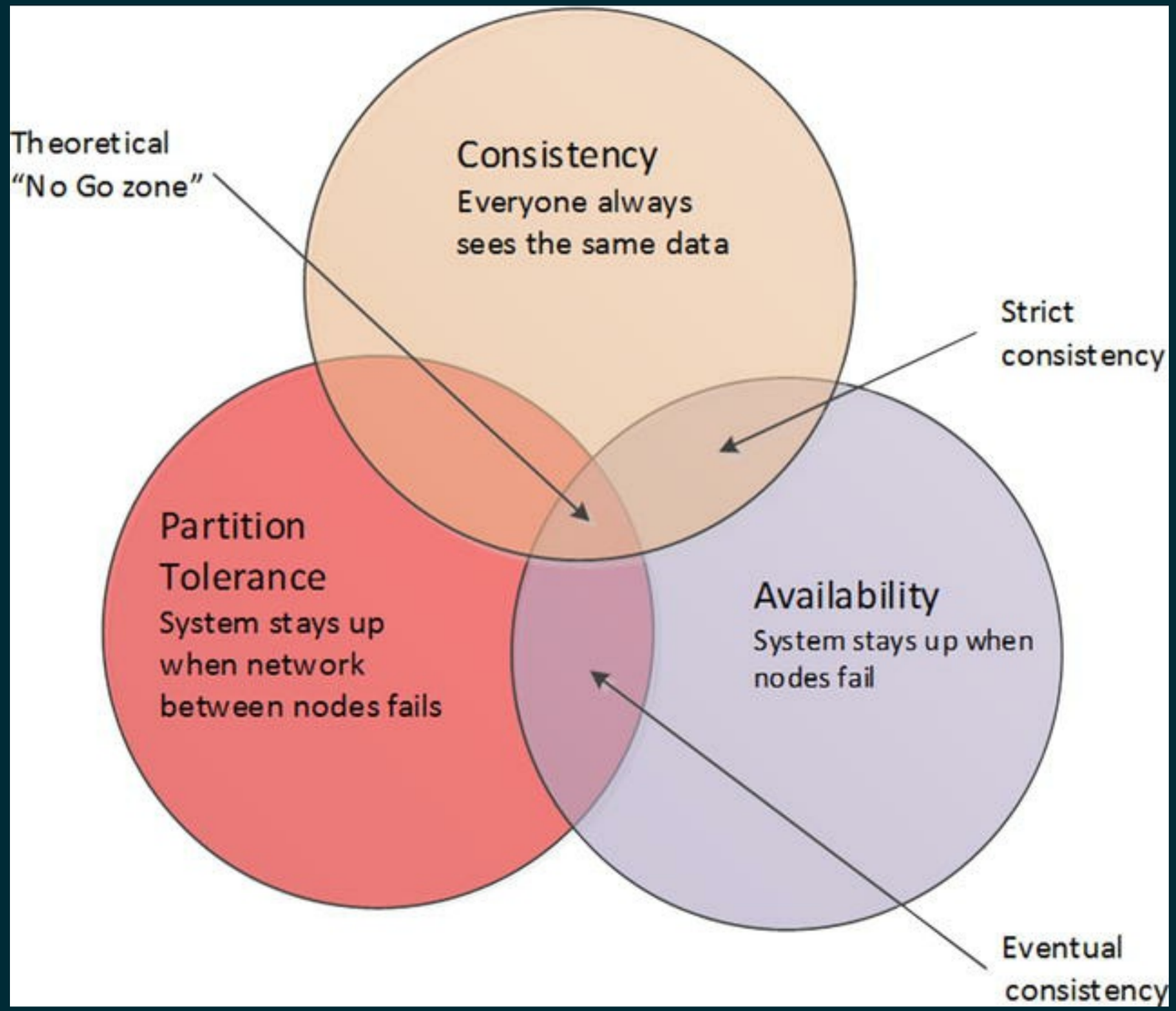
SCALABILITY

Scale Database Layer

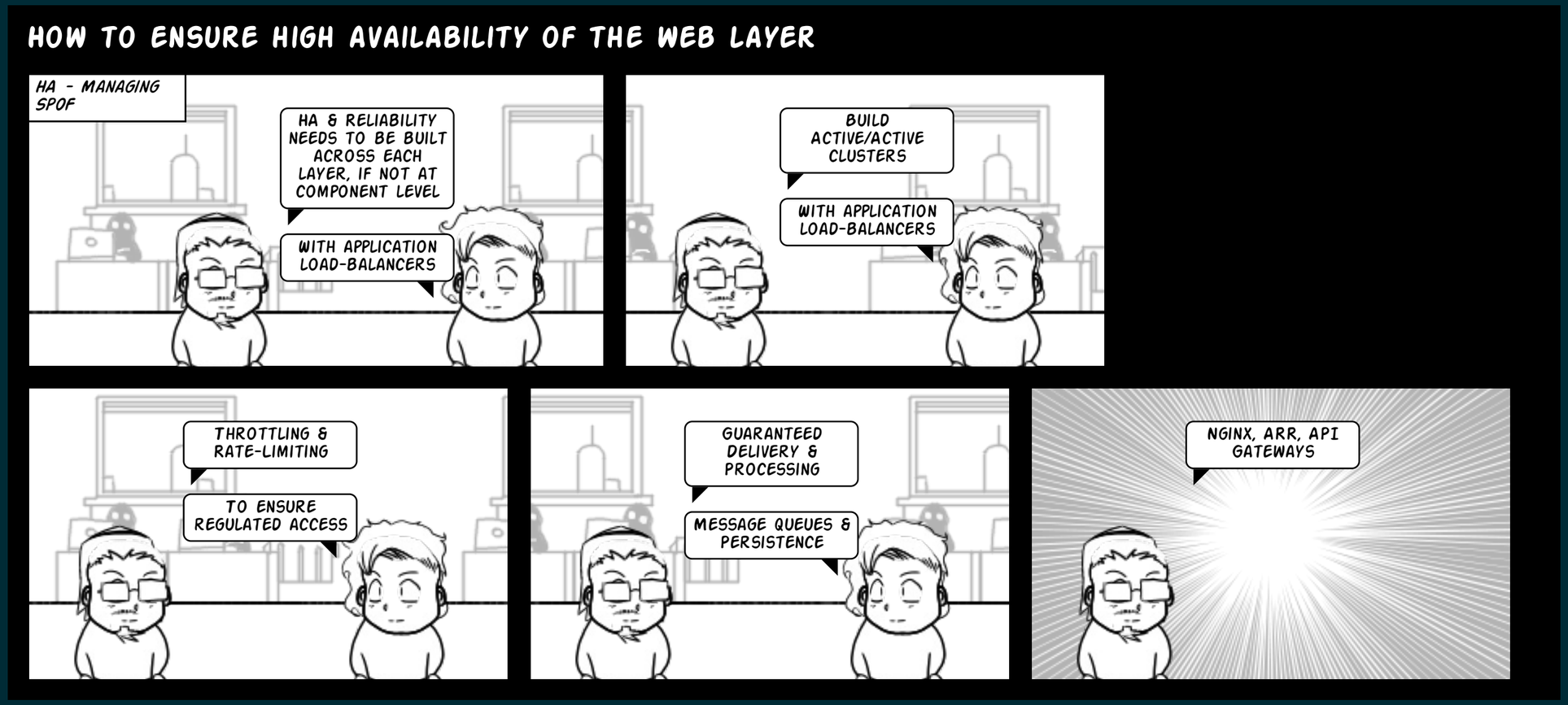


SCALABILITY

CAP Theorem Revisited

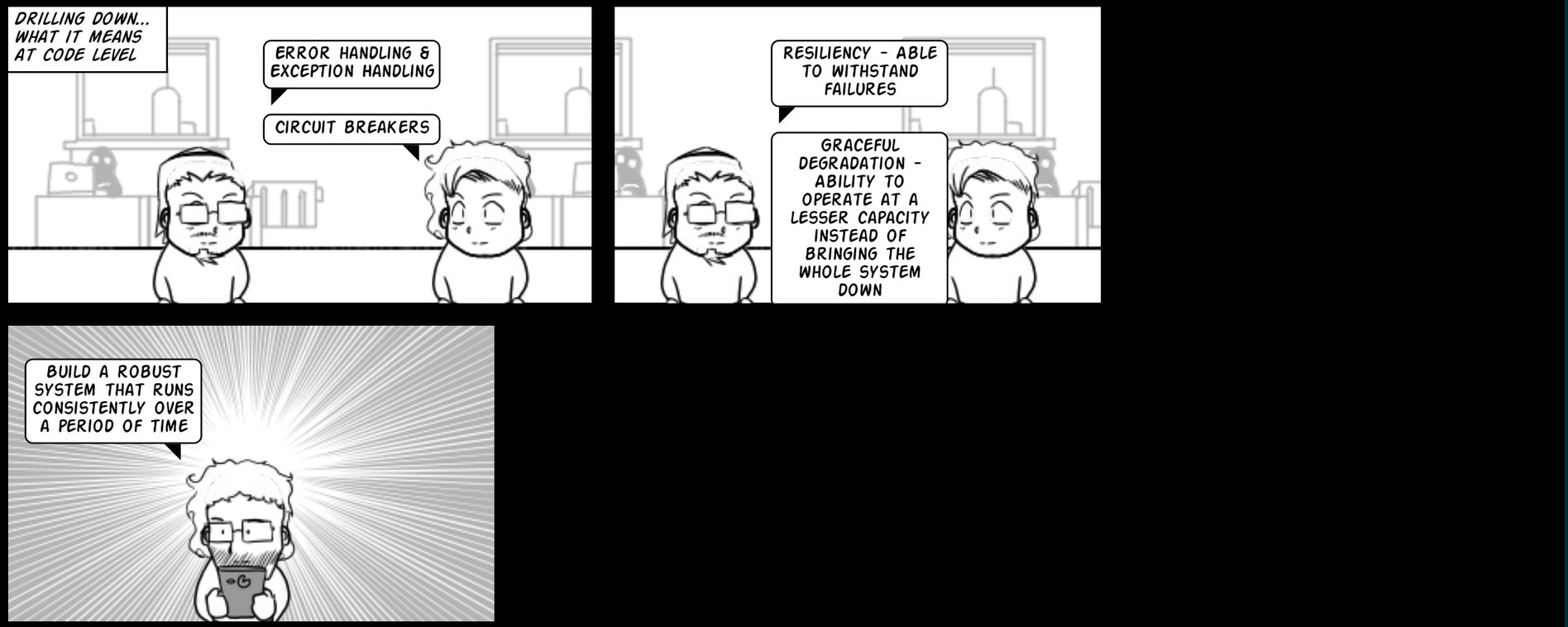


HA & RELIABILITY



HA & RELIABILITY

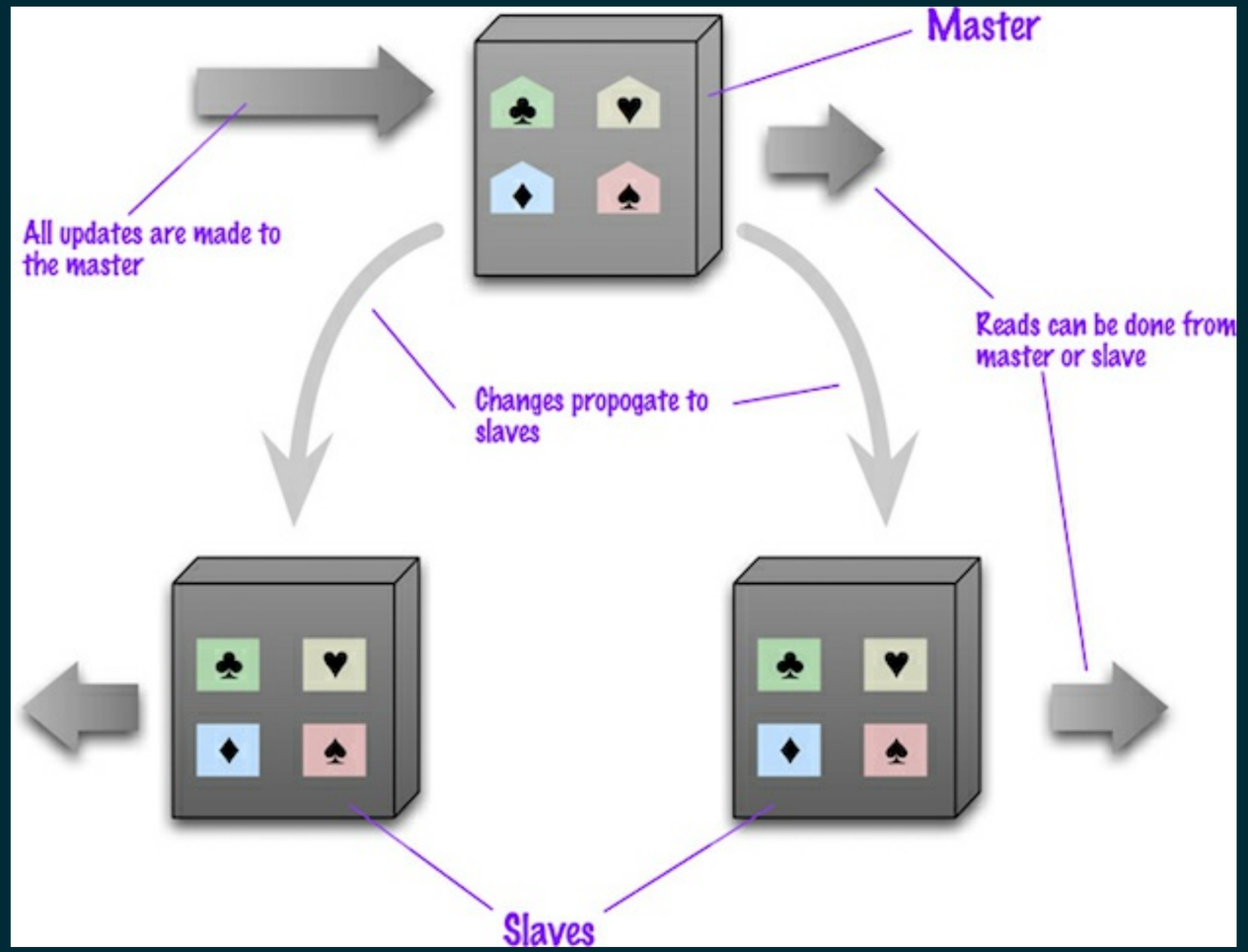
RELIABILITY, RESILIENCY & ROBUSTNESS



HA & RELIABILITY

Master Slave

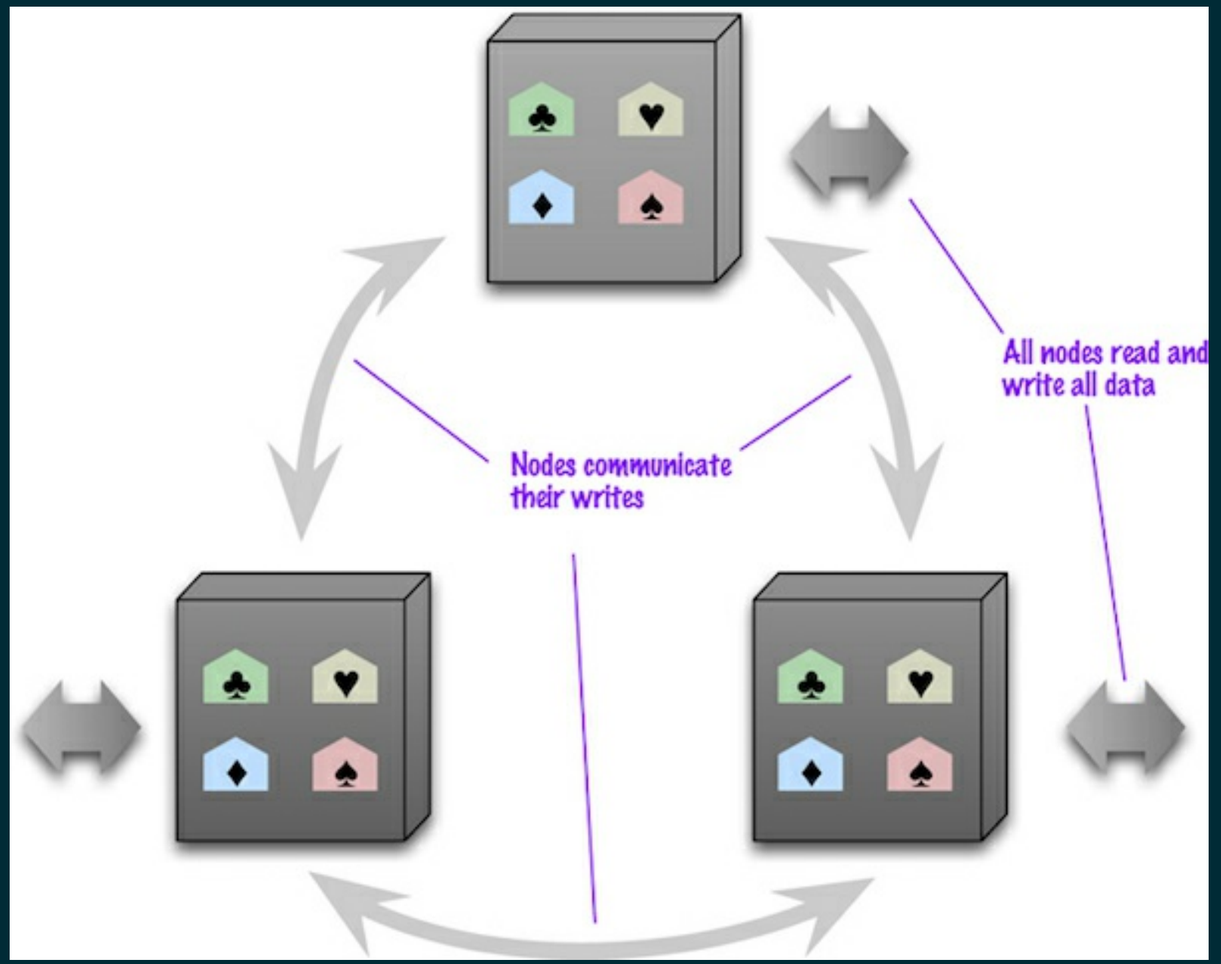
Master Slave replication



HA & RELIABILITY

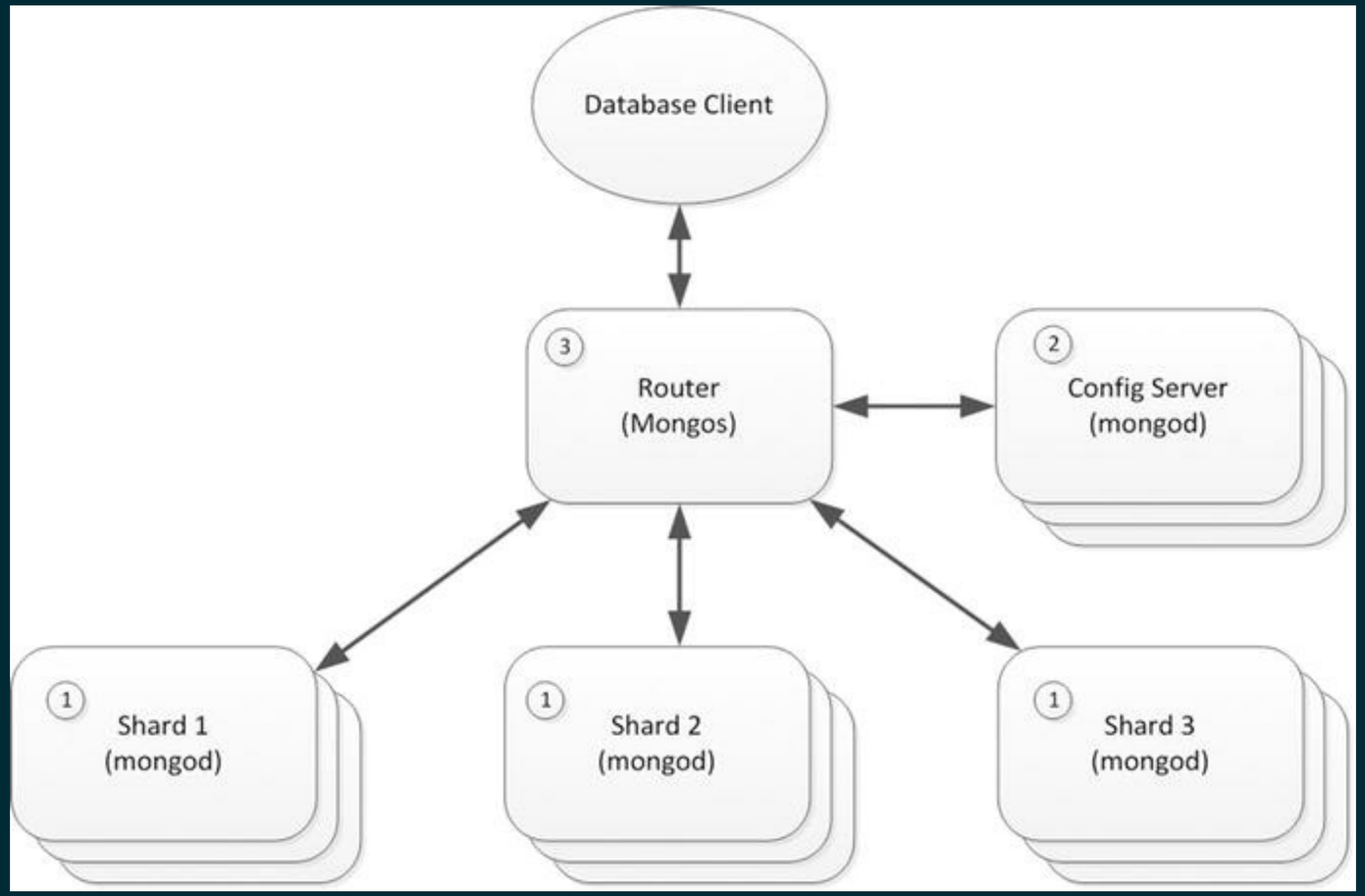
Master Slave

Peer to Peer replication



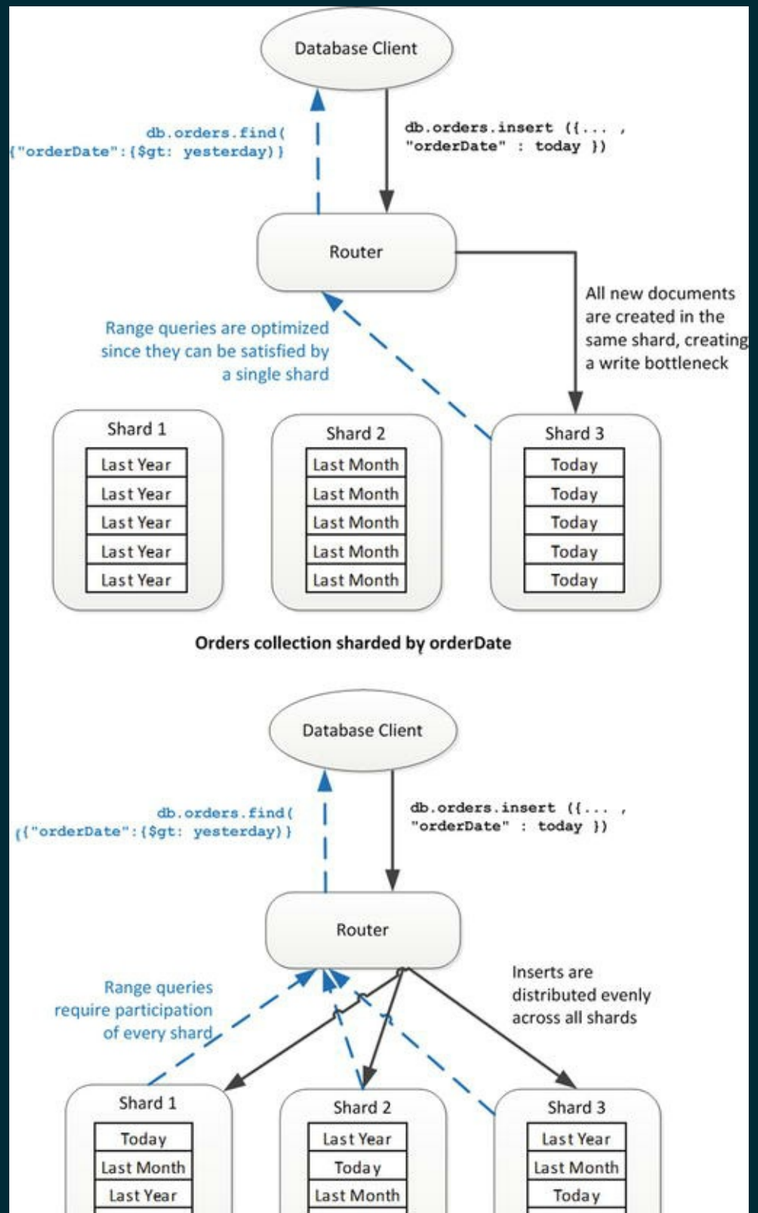
HA & RELIABILITY

Sharding of NoSQL Database (Mongo DB)



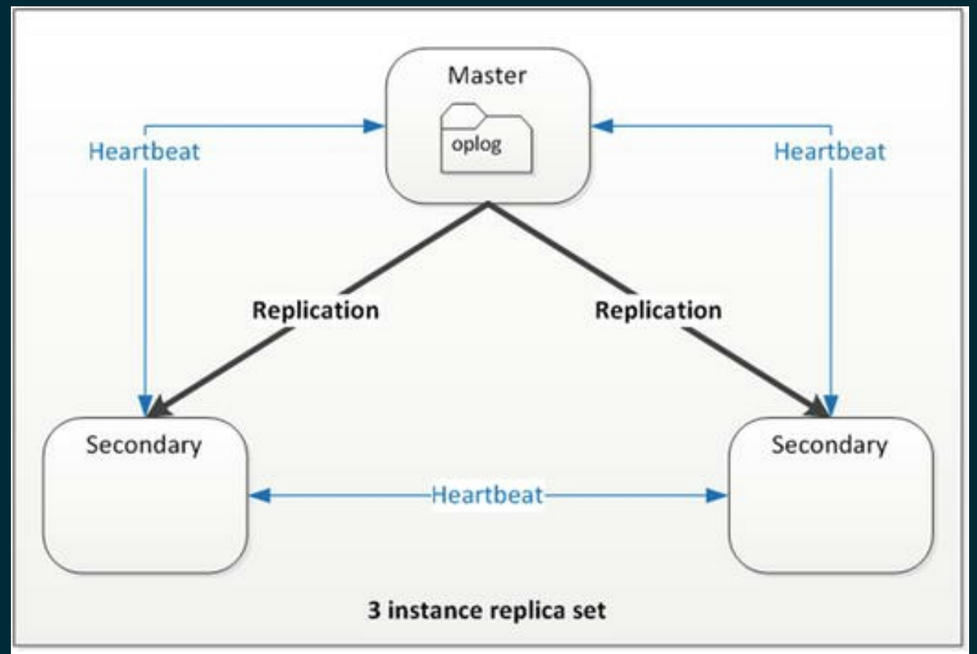
HA & RELIABILITY

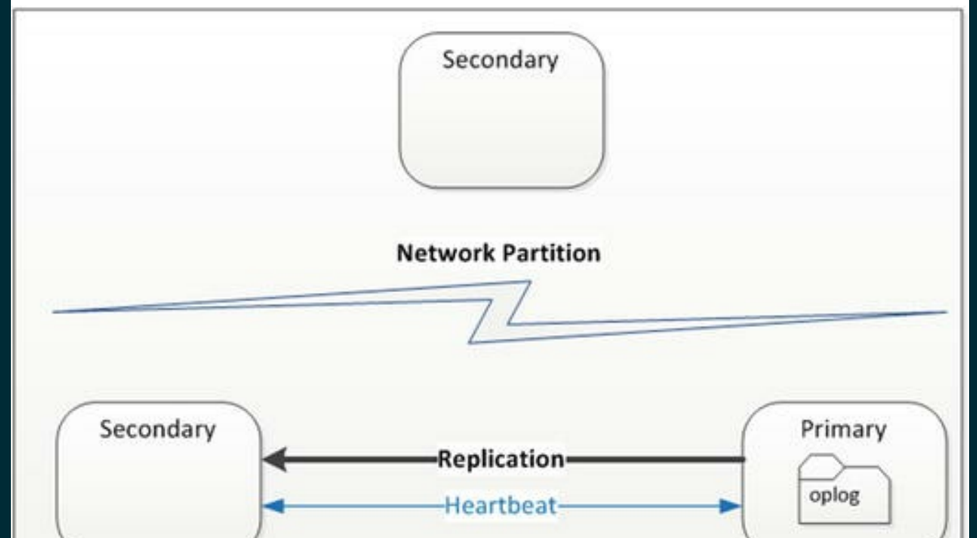
Sharding of NoSQL Database (Mongo DB .. Continued)



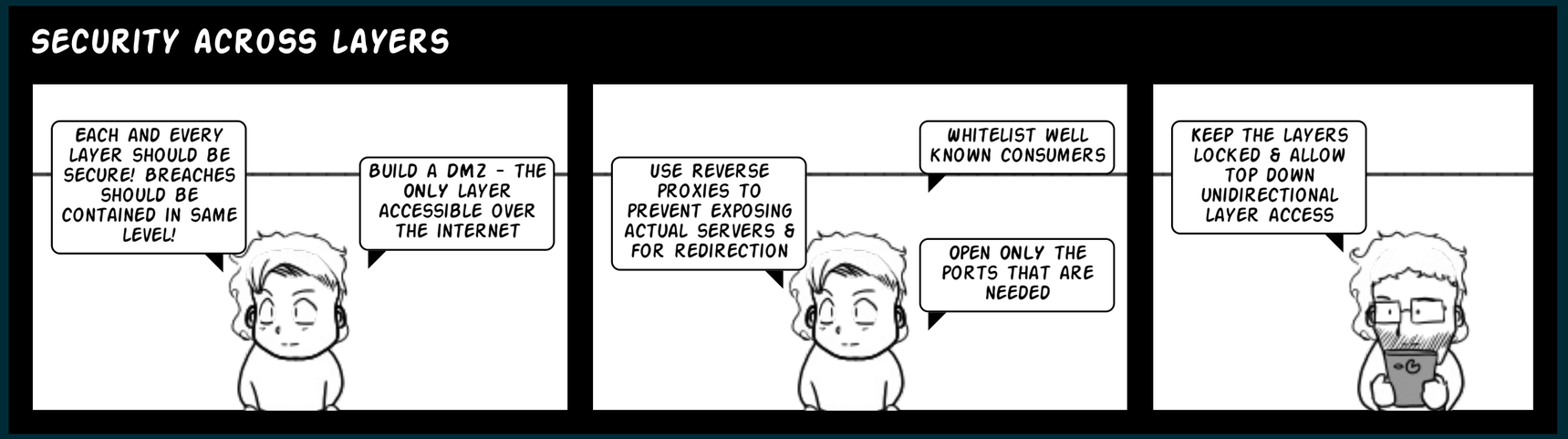
HA & RELIABILITY

Replication set



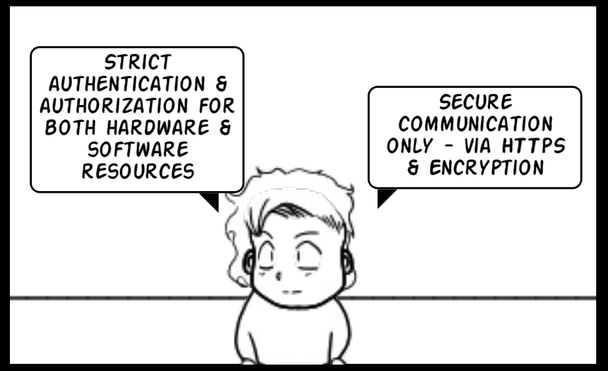


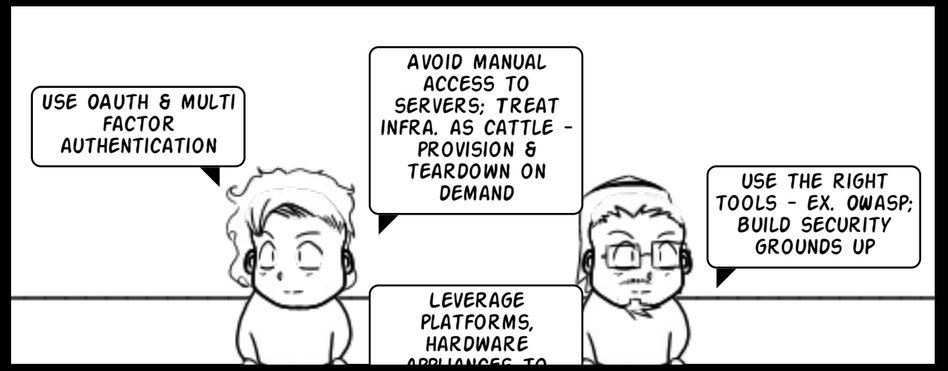
SECURING THE SYSTEM



SECURING THE SYSTEM

INTERNAL & EXTERNAL SECURITY



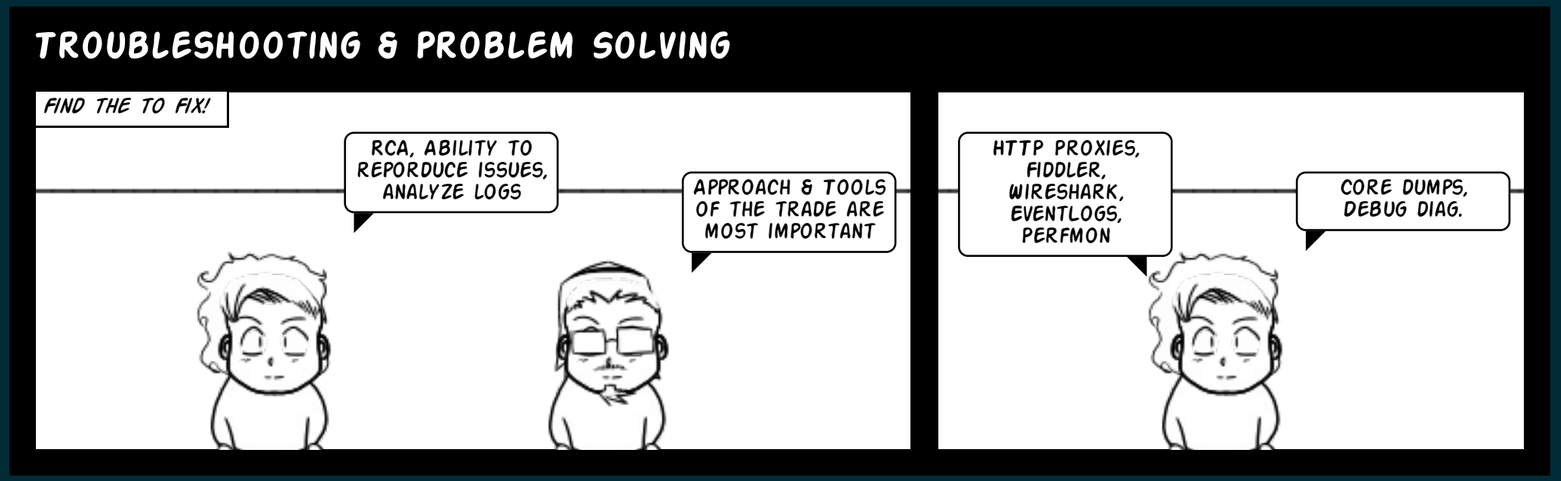


SECURING THE SYSTEM

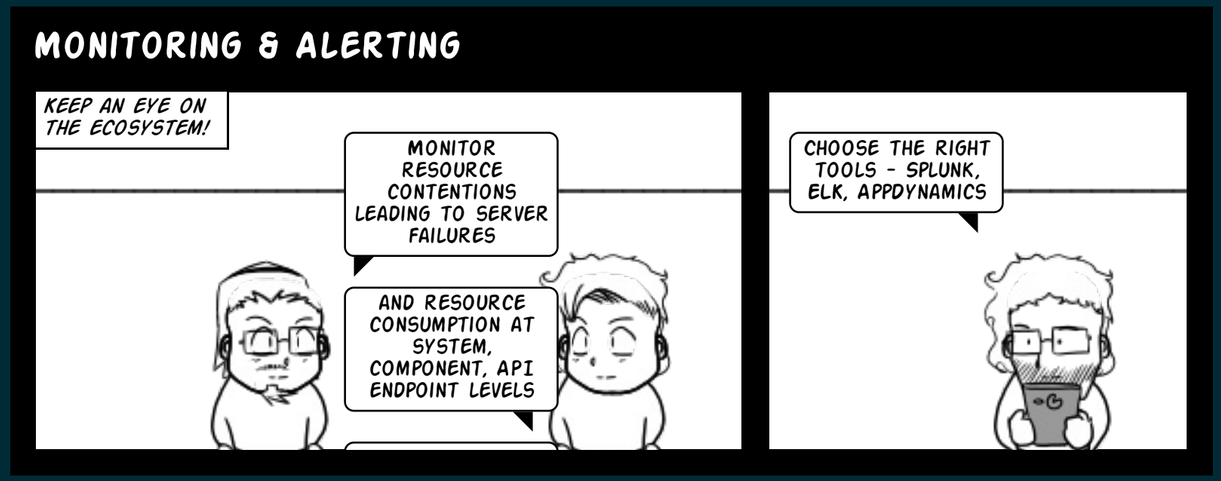
https://www.microsoft.com/en-us/download/confirmation.aspx?id=1330
http://azuresecurity.codeplex.com/

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!'
THANK YOU!


ARe We
THeRe
Yet?
A journey of a thousand miles
begins with a single step
... And lots of coffee!
Visit Jim Hunt at facebook.com/huntcartoons