Ardeim

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

Thanks to: Rashmi Subbanna (RashmiSubbanna@tesco.com)

Credits: revealjs, stripthis (kesiev)

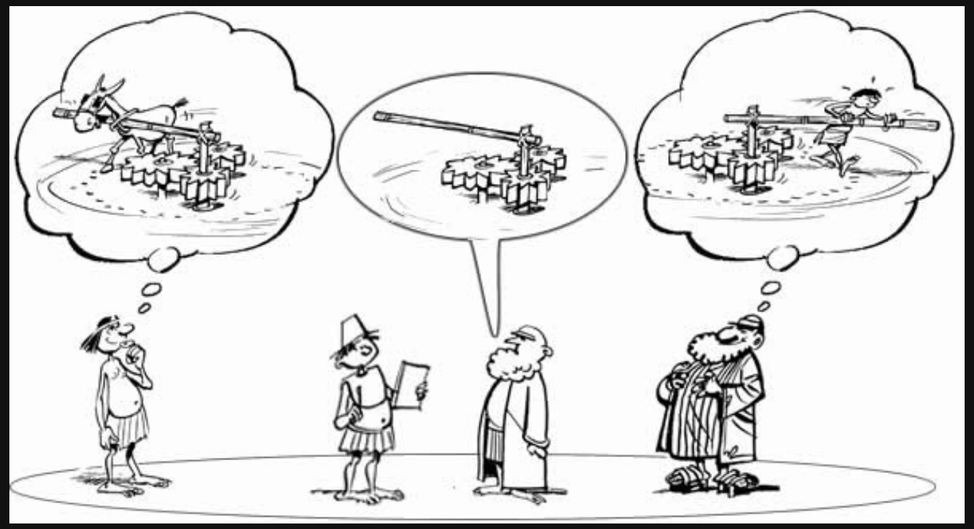
Heads UP!

Raise relevant questions to understand the business benefits!

- Be Open and Make No Assumptions -

Heads UP!

What you think is what you get!

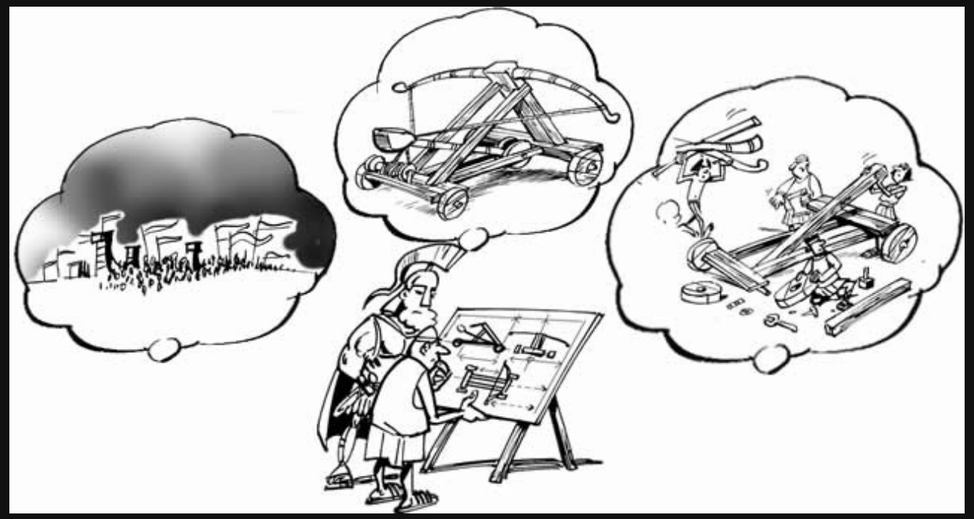


Specific & relevant details helps you think through a better solution!

- Stay Focused -

Heads UP!

Prespectives & Concerns Differ!



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

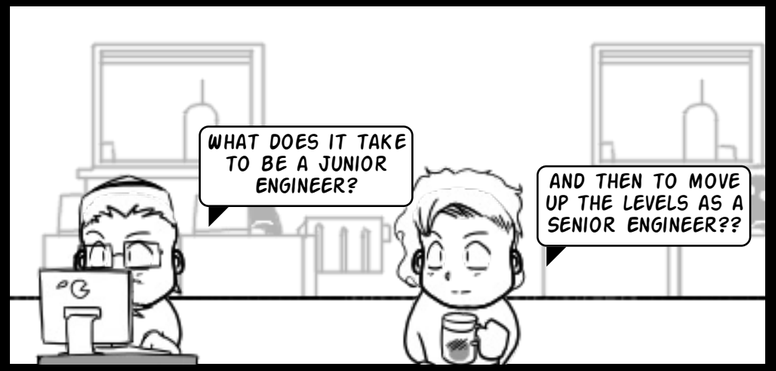
- Be an Enabler; Listen, Acknowledge & Accept Non Technical Needs! -

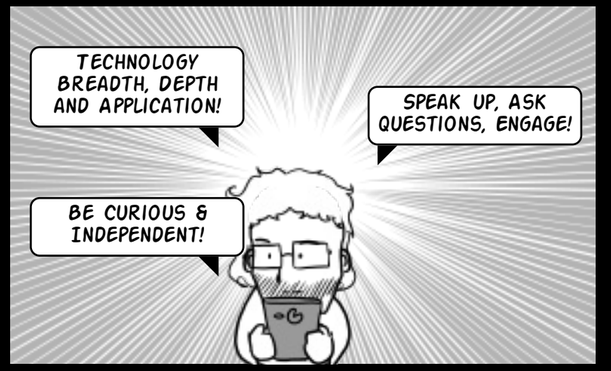
Context

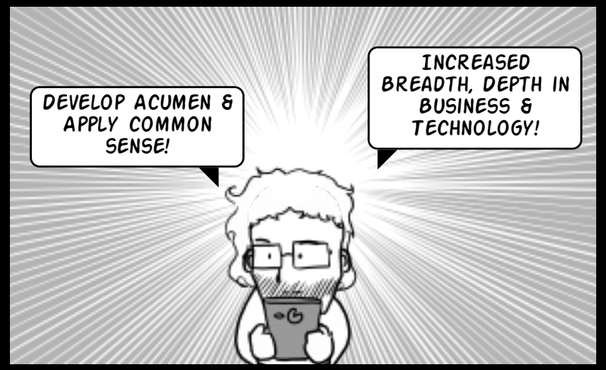

THE TEAM!
BIANCA*PROD. MANAGER
KESIEV*JR. ENGG.
LARA*SR. ENGG.
PETER*DIVINE VOICE!
ANNA*DEV. MANAGER

Context

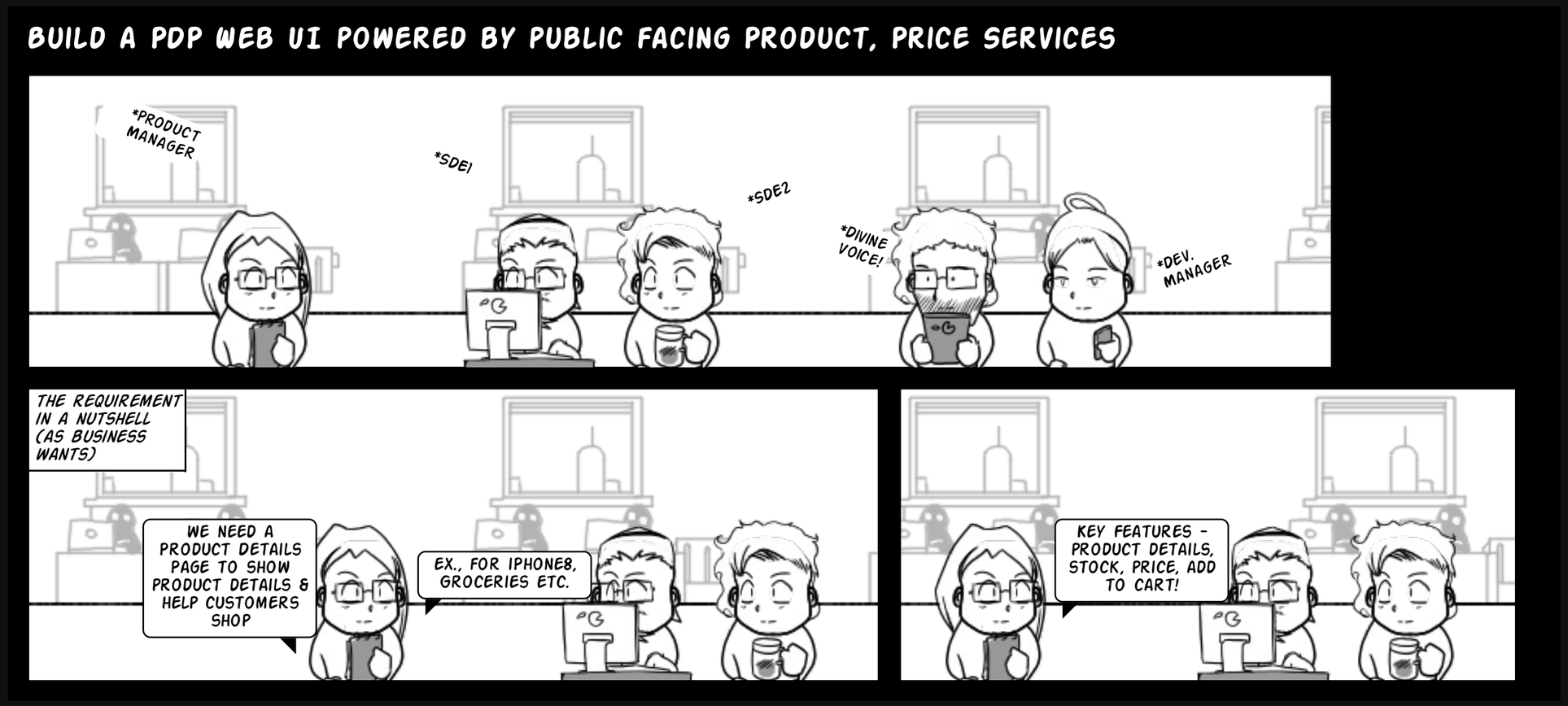
THE ENGINEER'S DILEMMA







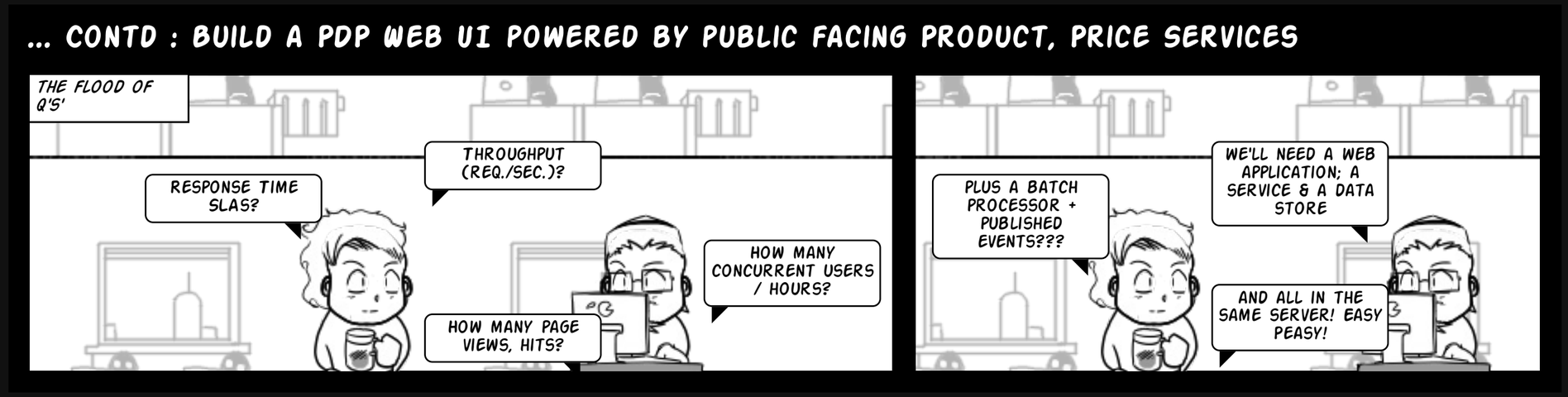
Context


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!

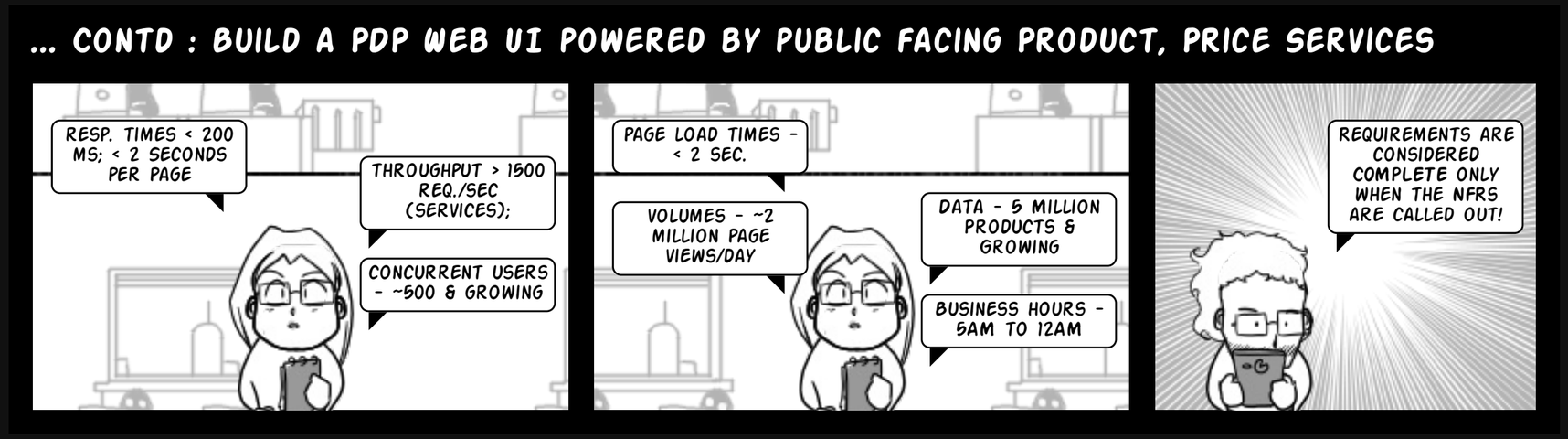
Context

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

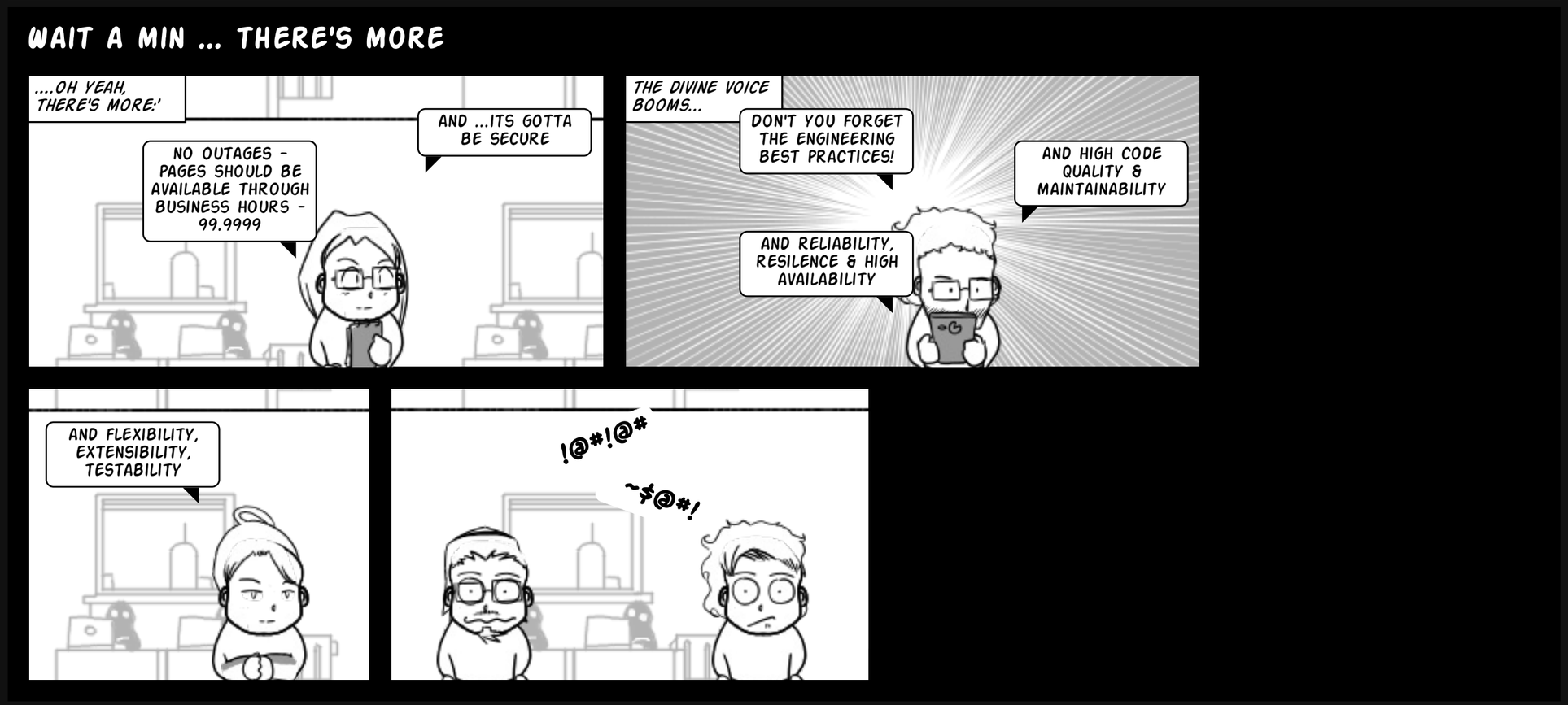
Context


... 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 CONCURRENT USERS / HOURS?
HOW MANY PAGE VIEWS, HITS?
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

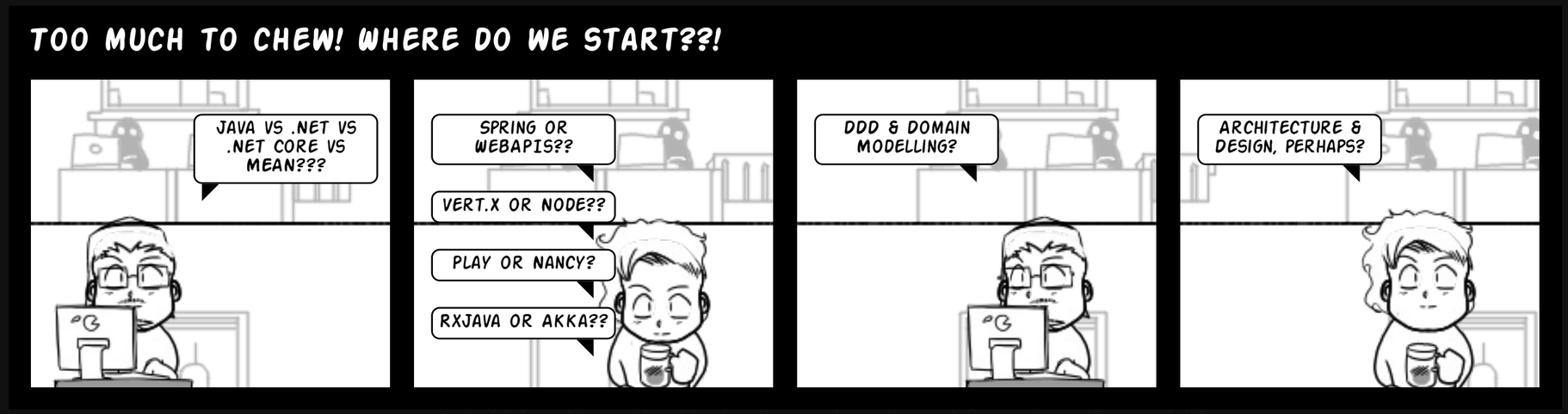

... CONTD : BUILD A PDP WEB UI POWERED BY PUBLIC FACING PRODUCT, PRICE SERVICES
RESP. TIMES < 200 MS; < 2 SECONDS PER PAGE
THROUGHPUT > 1500 REQ./SEC (SERVICES);
CONCURRENT USERS - ~500 & GROWING
PAGE LOAD TIMES - < 2 SEC.
VOLUMES - ~2 MILLION PAGE VIEWS/DAY
DATA - 5 MILLION PRODUCTS & GROWING
BUSINESS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

Context



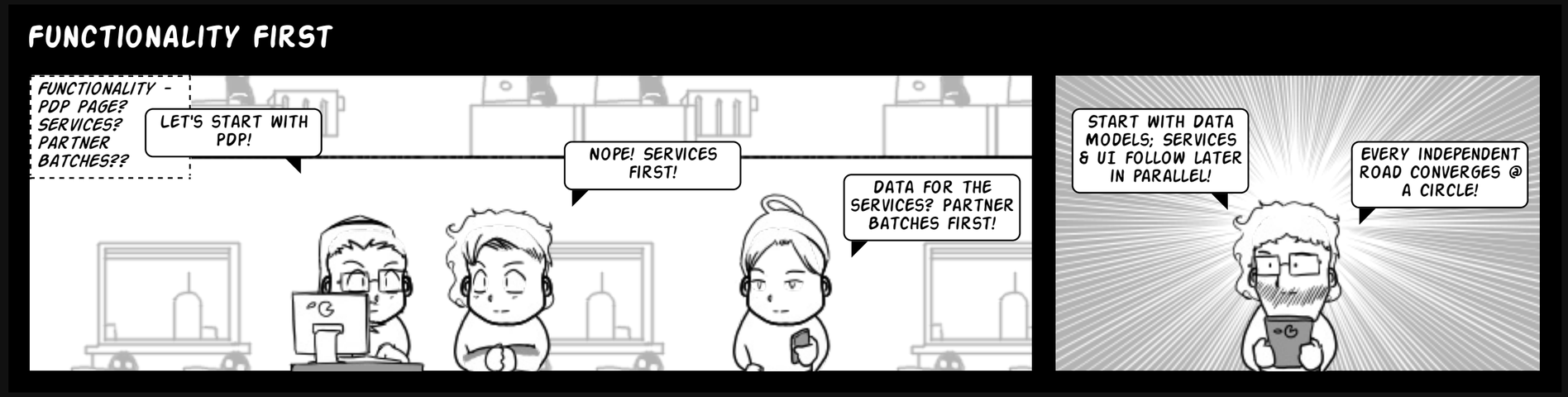
Whew! So, What Next?

What next!?

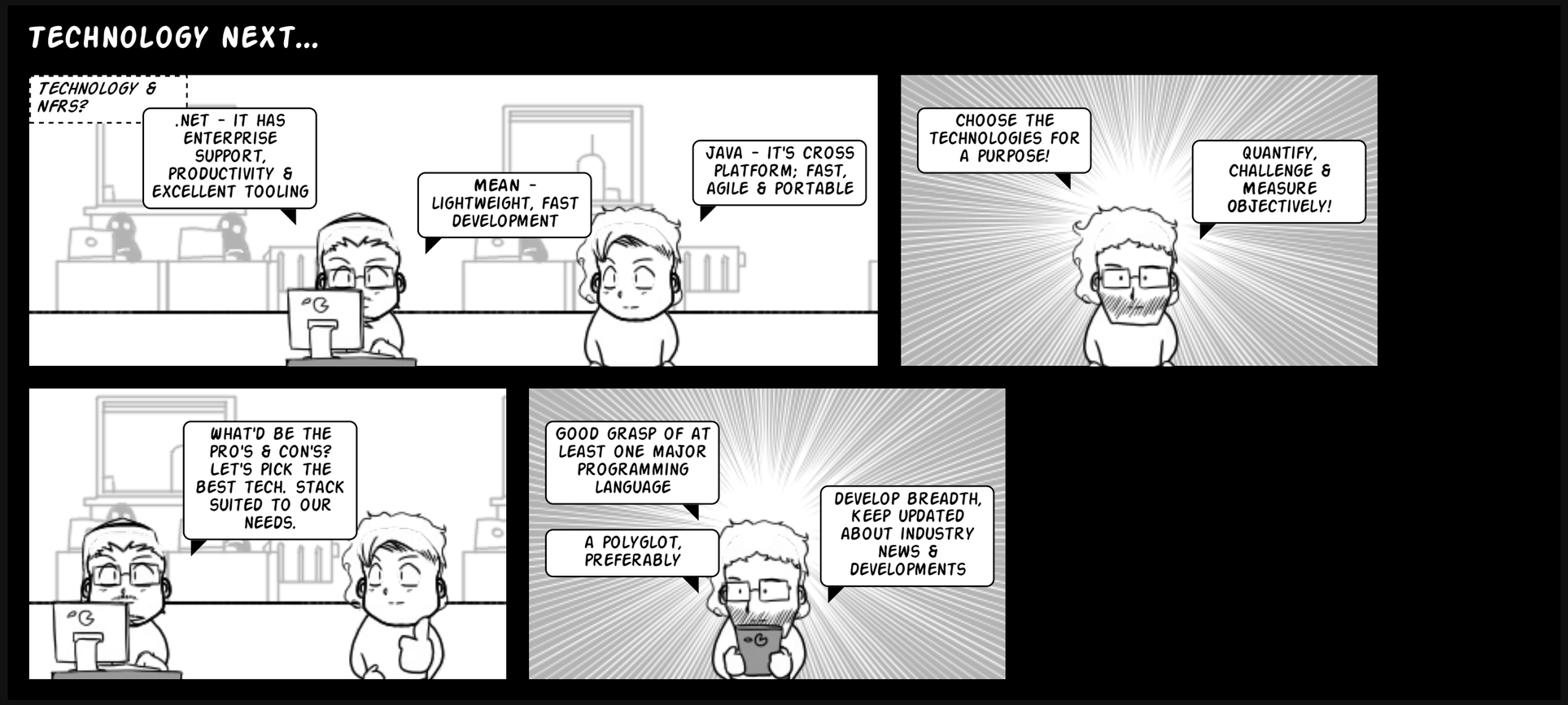


- Build Functionality first, then Iterate addressing the NFRs -

What next!?


FUNCTIONALITY FIRST
FUNCTIONALITY - PDP PAGE? SERVICES? PARTNER BATCHES??
LET'S START WITH PDP!
NOPE! SERVICES FIRST!
DATA FOR THE SERVICES? PARTNER BATCHES FIRST!
START WITH DATA MODELS; SERVICES & UI FOLLOW LATER IN PARALLEL!
EVERY INDEPENDENT ROAD CONVERGES @ A CIRCLE!

What next!?


TECHNOLOGY NEXT...
TECHNOLOGY & NFRS?
.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'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!?

- Fret Not! Help is not far! -

What next!?

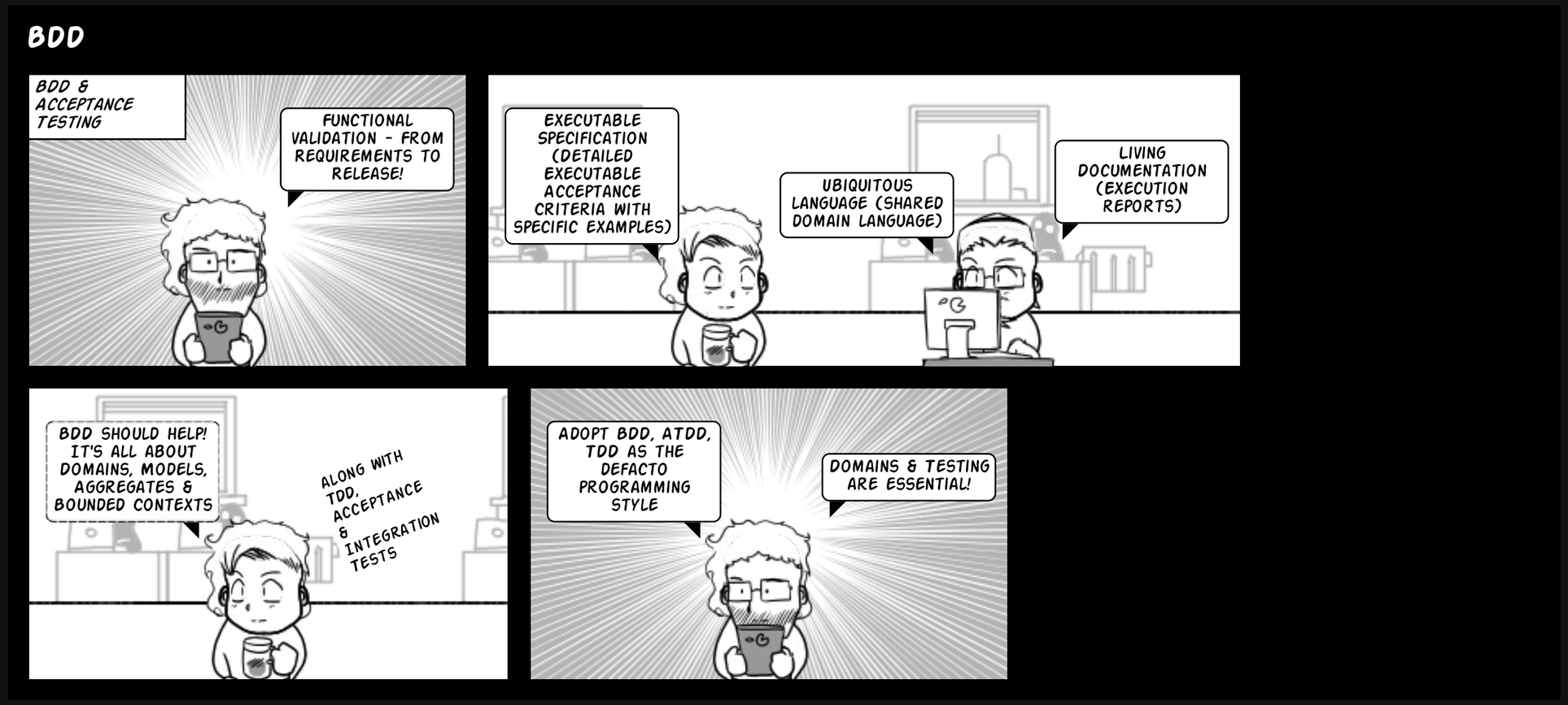
Data first, Design & Implementation follows next...

Core Concepts
- TDD, BDD, DDD

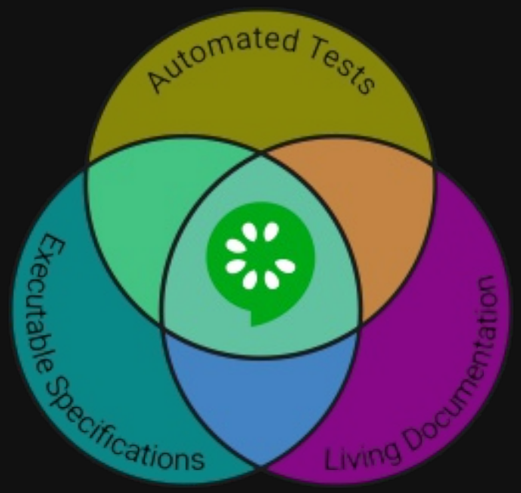
Architecture & Design
- Architectural Styles & Views
- Technology Selection (How & Why)
- Polyglot DataStores
- NFRs & Tradeoffs

Engineering
- Best Practices
- Design Patterns
- Design Principles (SOLID)
- Code Smells
- CI-CD & DevOPS first culture
- Programming Paradigms (Functional, Imperative)

Core Concepts ...

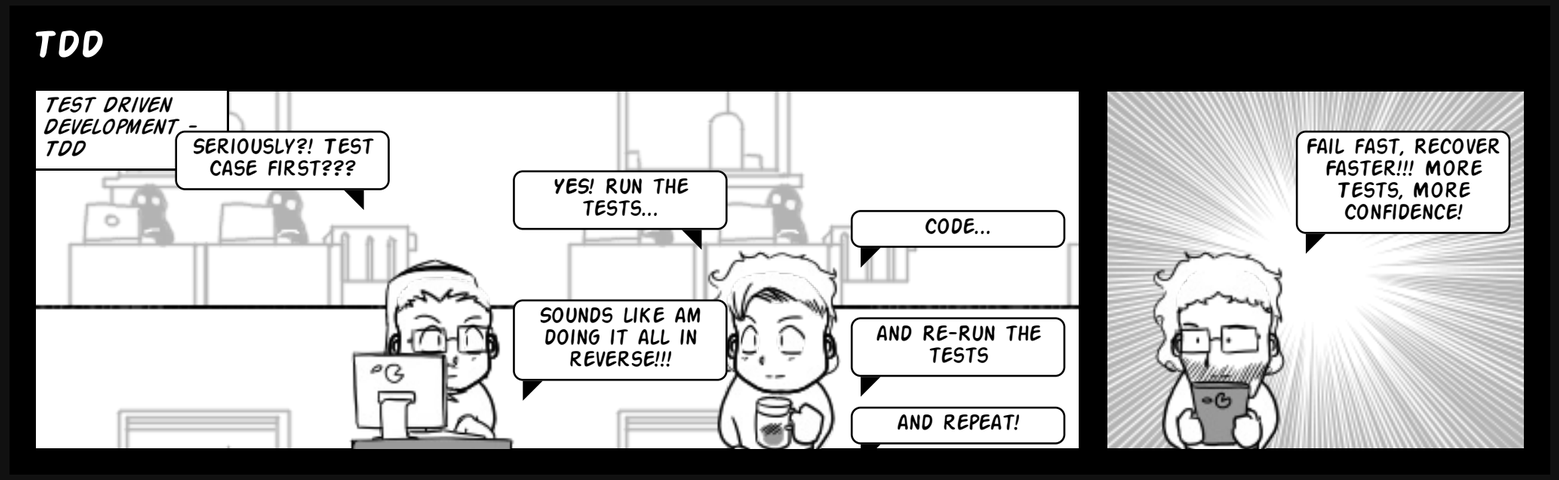


Core Concepts ...



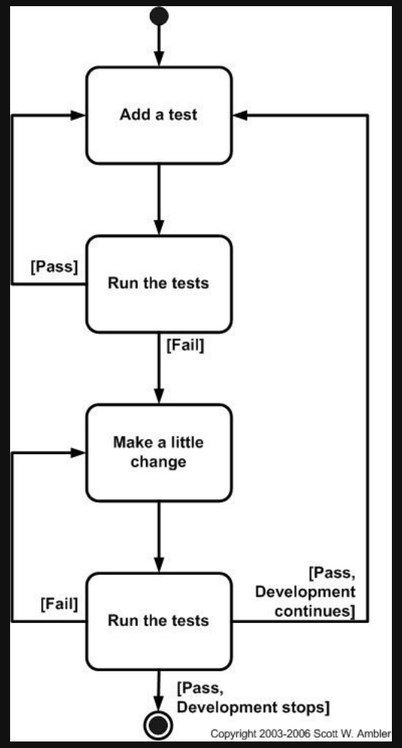
Benefits of BDD
BDD Frameworks
Book: BDD in Action
BDD - Introduction from Dan North

Core Concepts ...

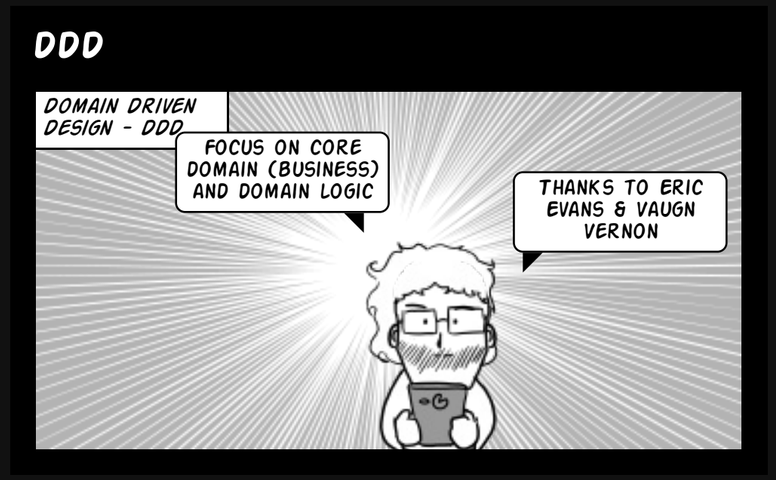


TDD Example
Introduction to TDD

Core Concepts ...

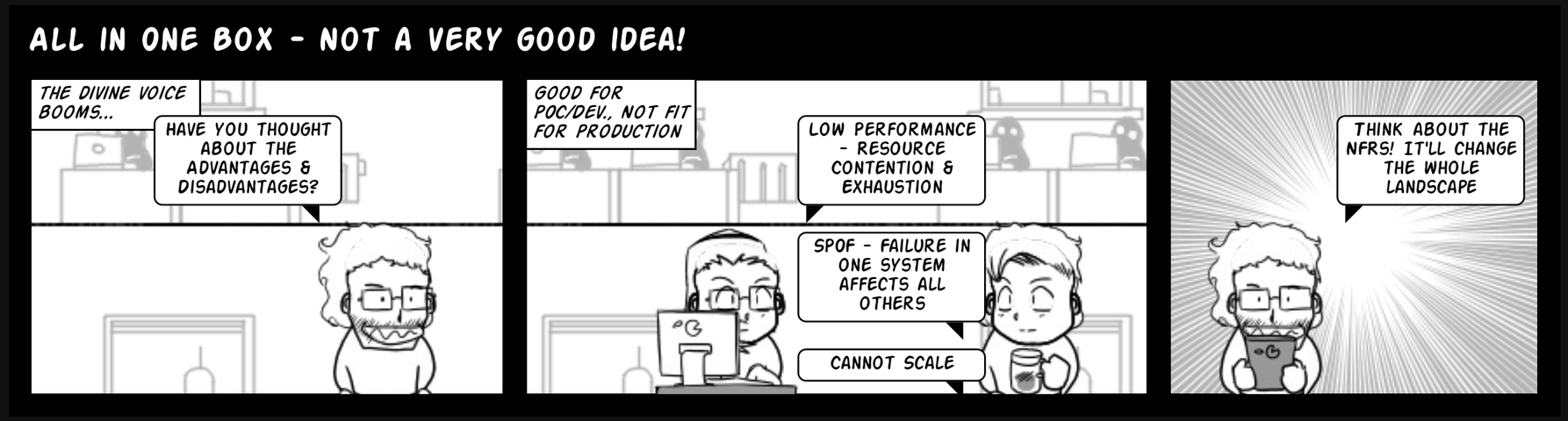


Core Concepts ...



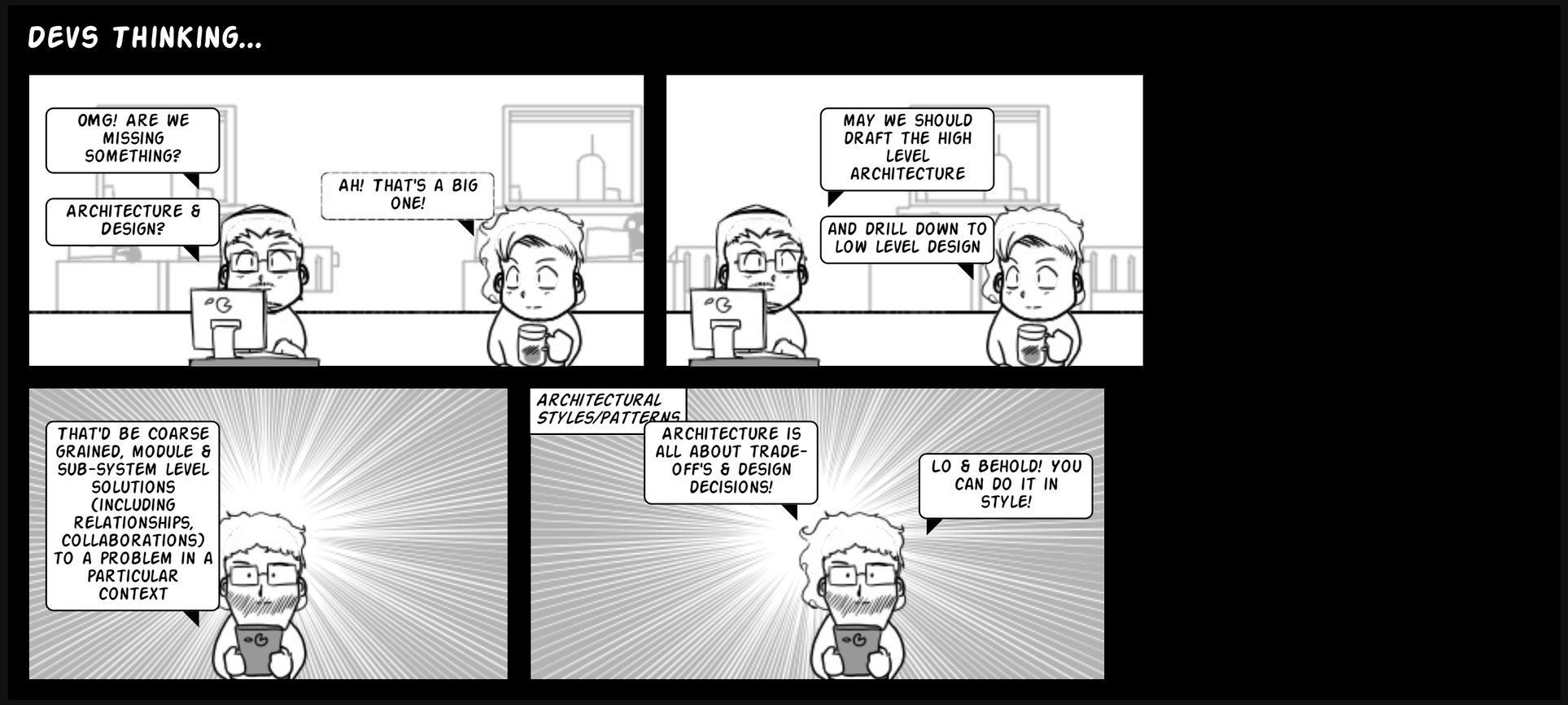
CQRS & Domain Driven Design
Awesome DDD - Nick Chamberlain
DDD Example
DDD Books
Domain Driven Design - Reference
The Ideal Domain-Driven Design Aggregate Store?
DDD & MicroServices

Architecture & Design ...

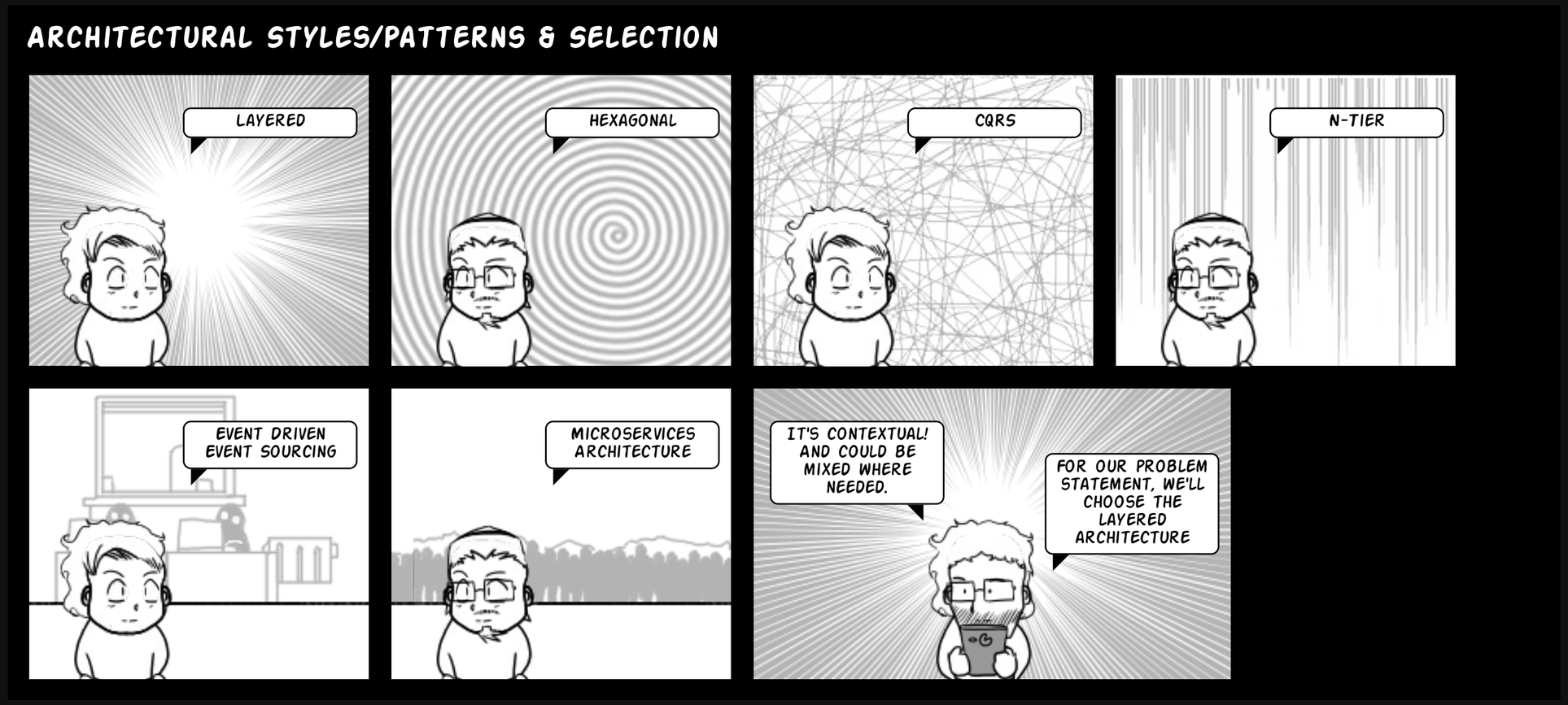


NFRs / QA (Quality Attributes)

Architecture & Design ...

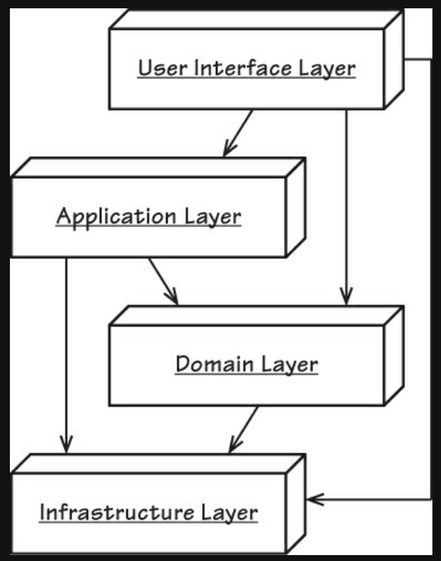


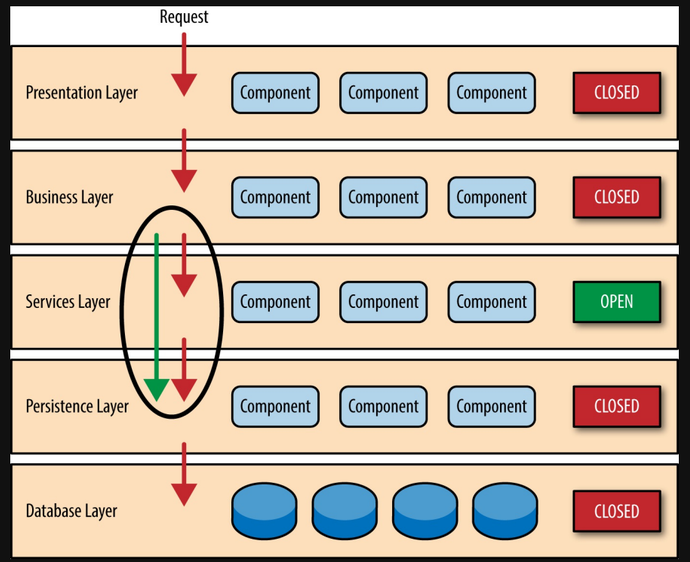
Architecture & Design ...

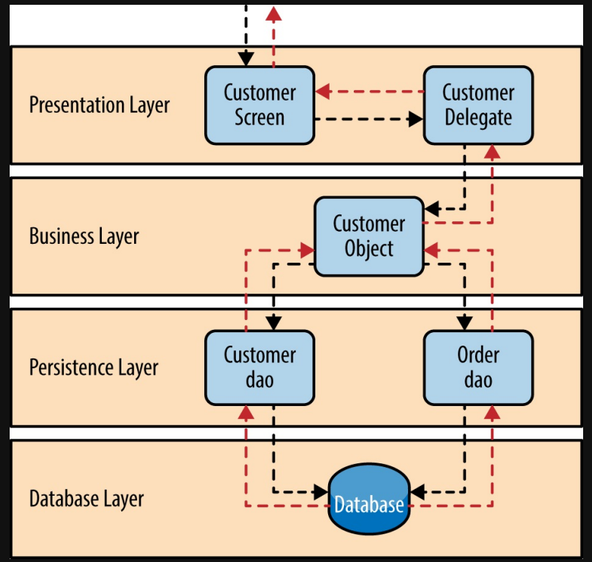


Architecture & Design ...

Layered Architecture





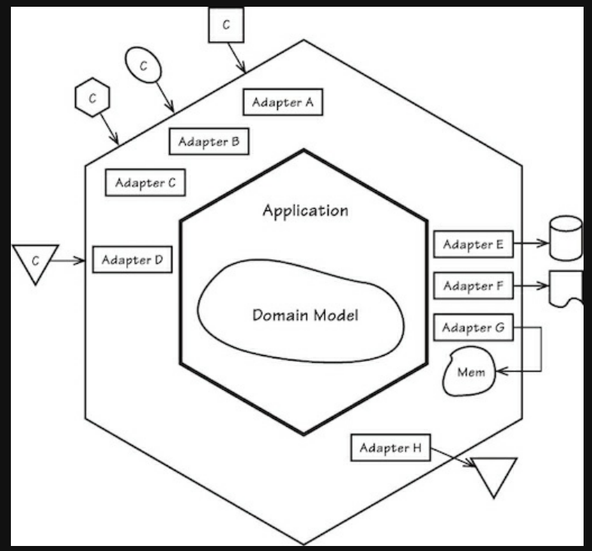


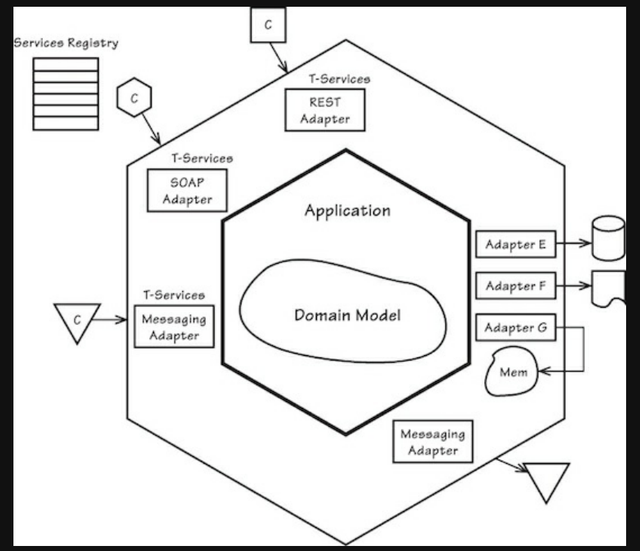
- Separation of Concerns - Simple Logical Decomposition, Top Down with Isolation -

Layered Architecture
N-Tier

Architecture & Design ...

Hexagonal Architecture



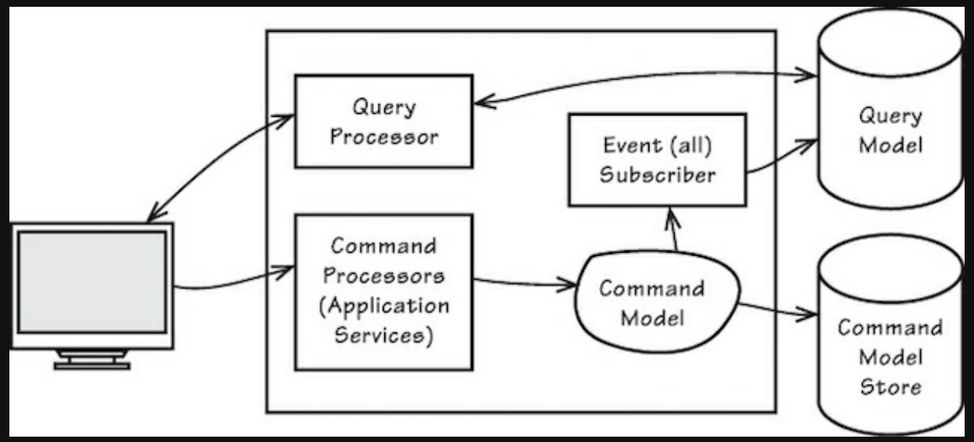


-Allow an application to equally be driven by users, programs, automated test or batch scripts, and to be developed and tested in isolation from its eventual run-time devices and databases-

Hexagonal Architecture
Hexagonal Architecture FAQ
Hexagonal Architecture Example #1
Hexagonal Architecture Example #2

Architecture & Design ...

CQRS

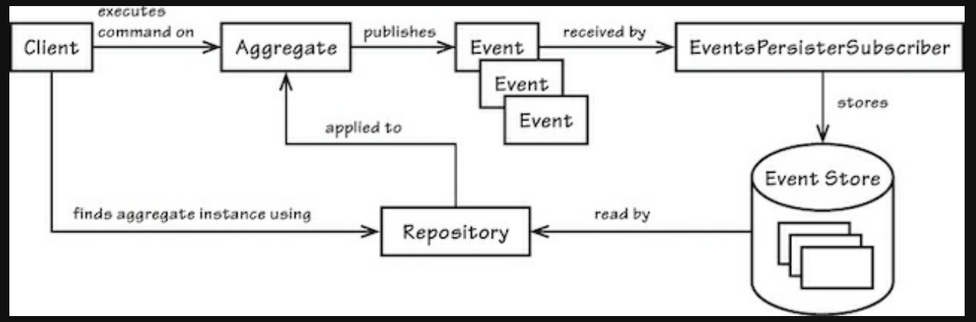


- Separate FLow for Queries (Reads) and Create (Writes) -

CQRS

Architecture & Design ...

Event Sourcing

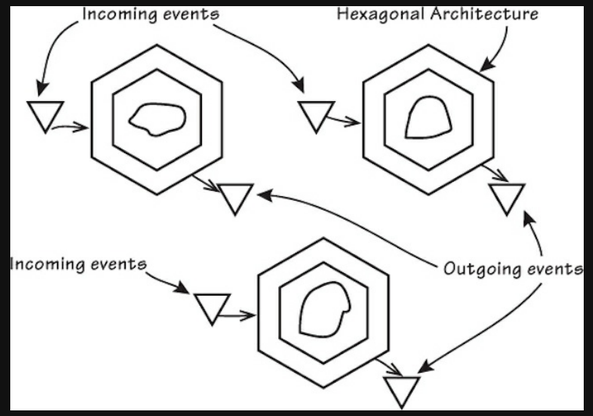


- Events are Immutable! There are no Updates + Deletes. Changes in state are stored as new events -

Event Sourcing Concept - Martin Fowler
Event Sourcing Example #1
Event Sourcing Example #2

Architecture & Design ...

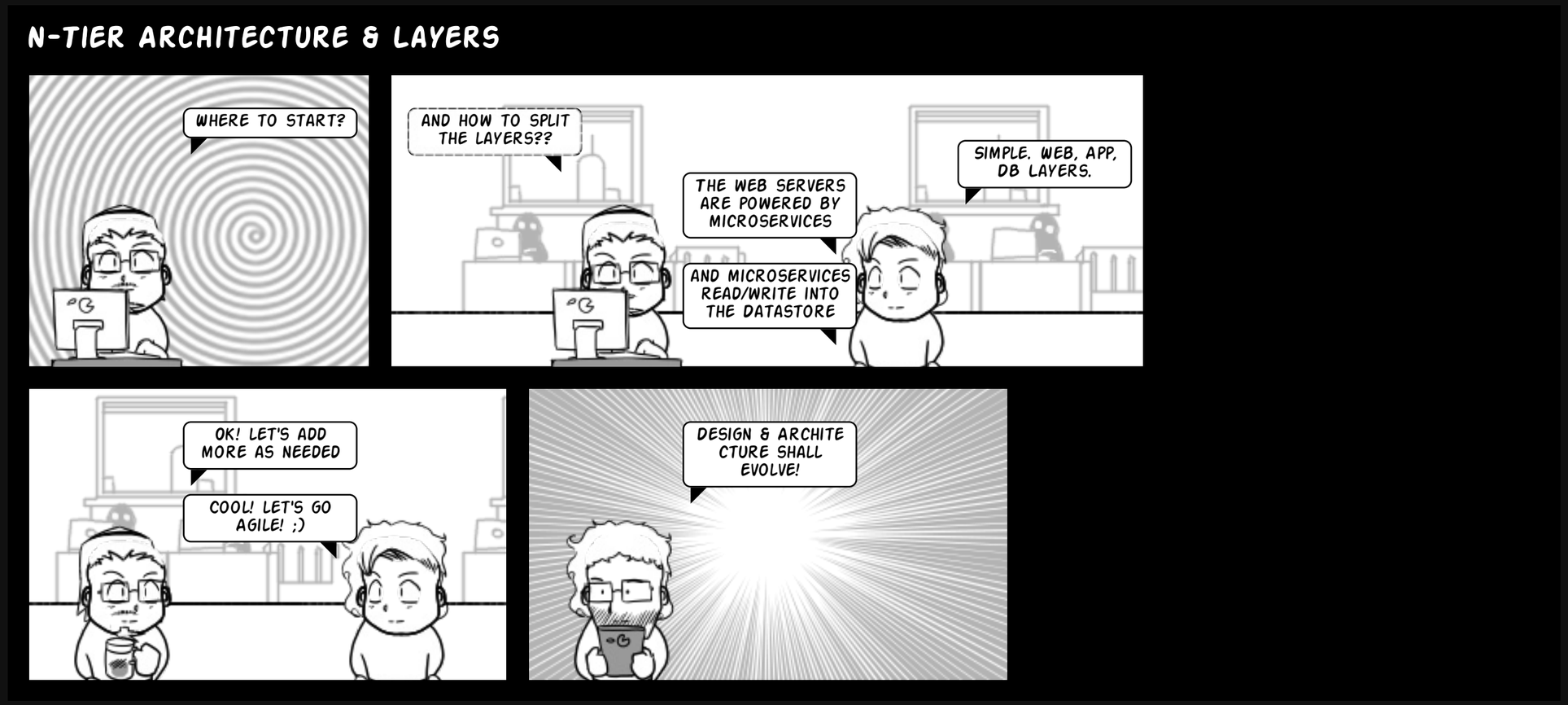
Event Driven



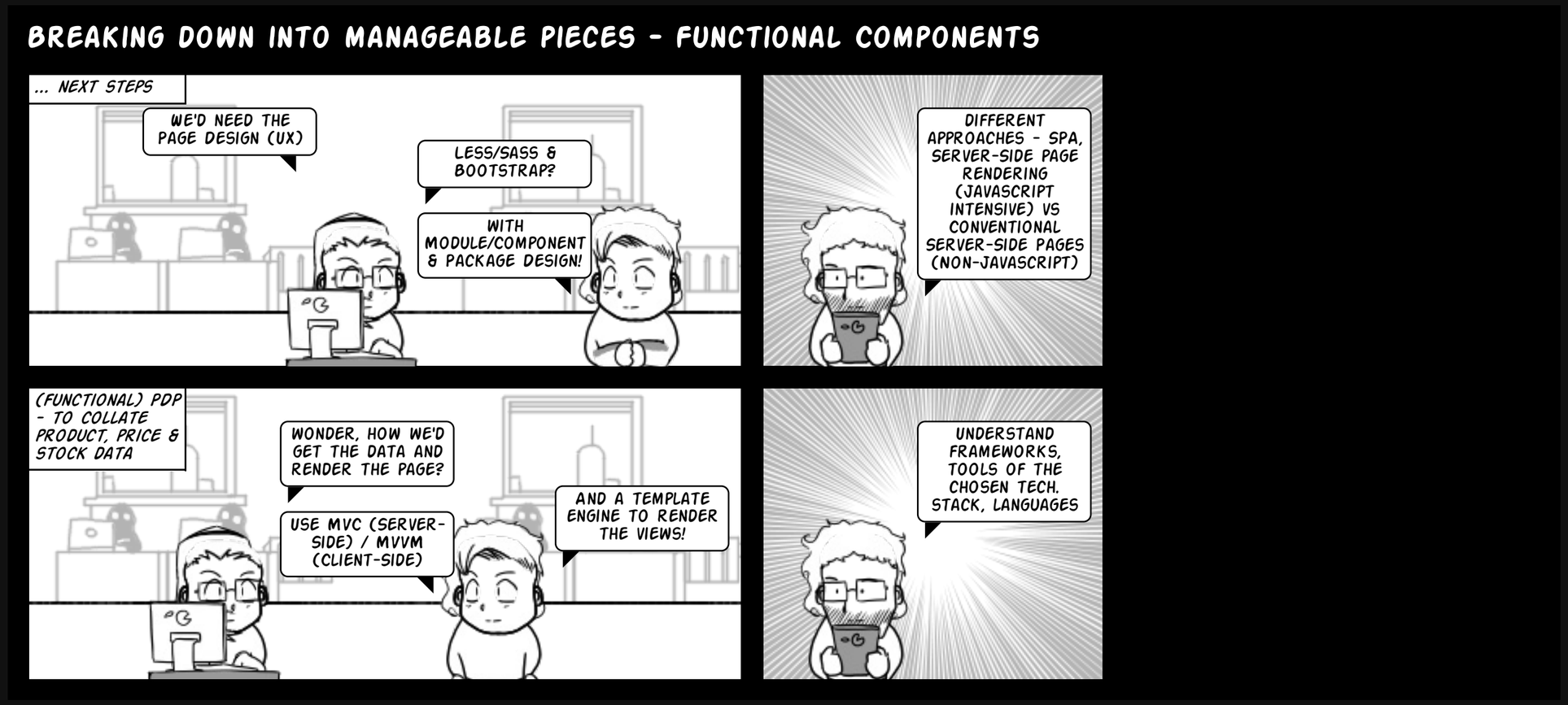
- Different Systems integrate using Domain Events -

Event Driven Architecture
Event Driven Data Management for Microservices
Event Driven Architecture - Example #1
Event Driven Architecture - Example #2 - Saga Pattern

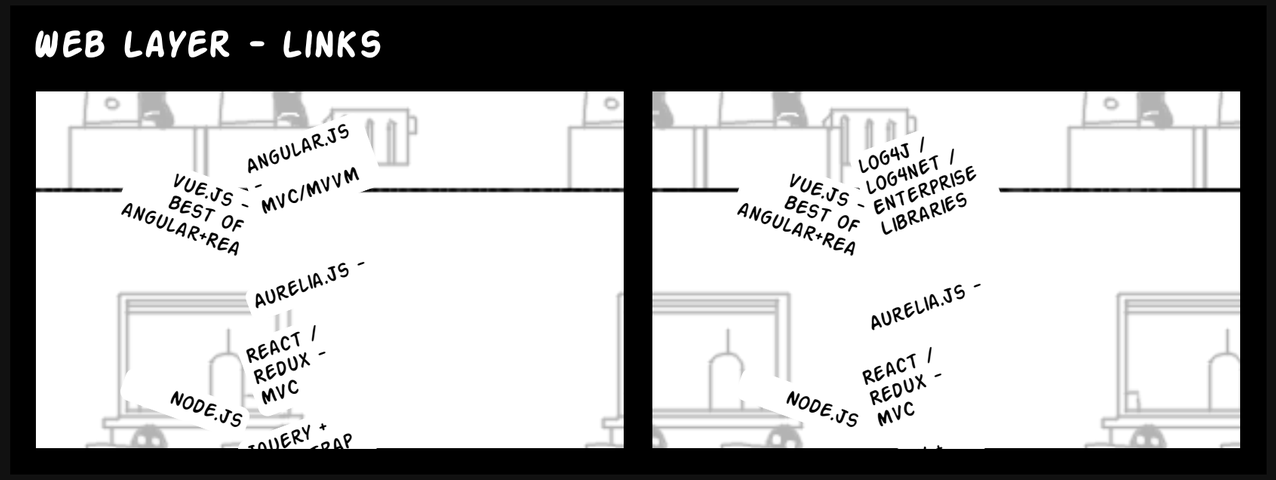
Architecture & Design ...



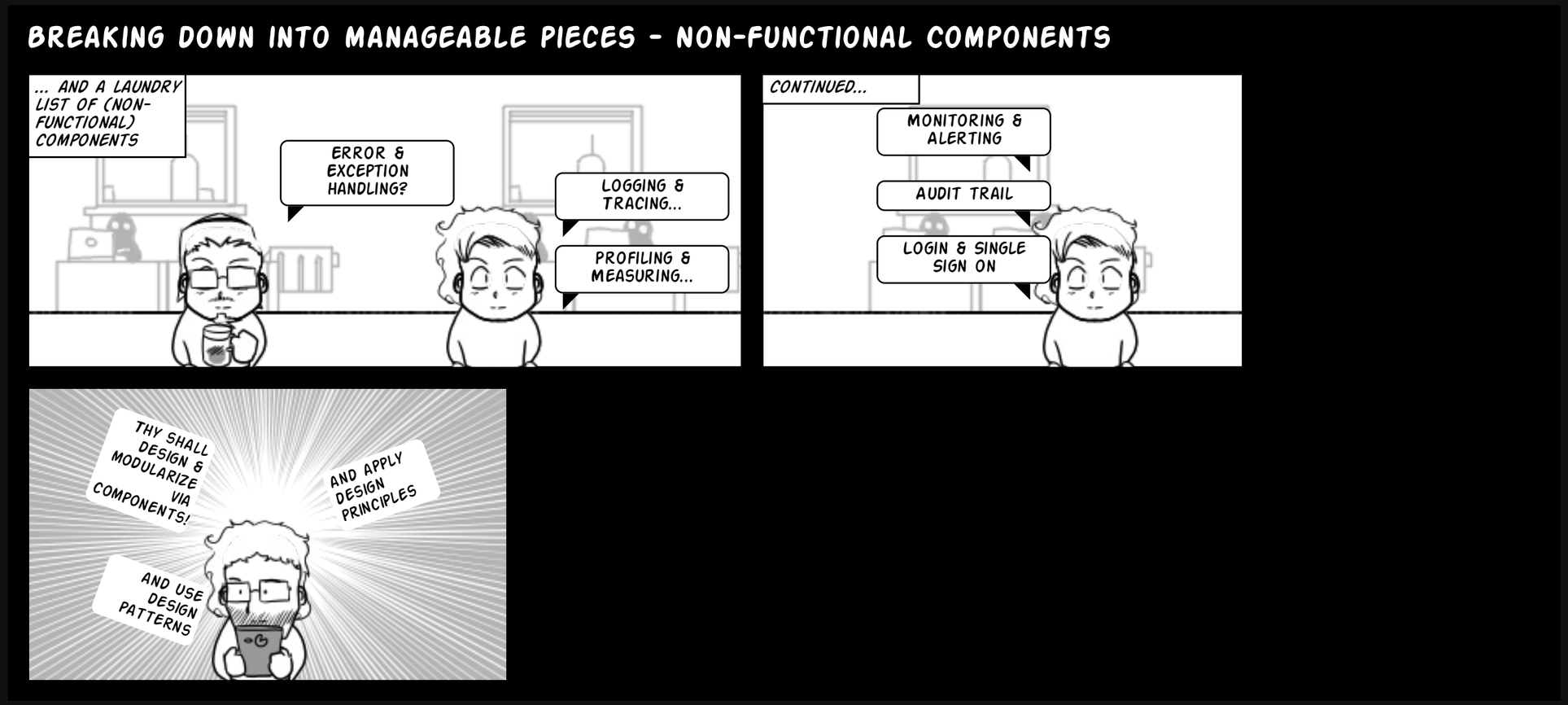
Architecture & Design ...



Architecture & Design ...



Architecture & Design ...

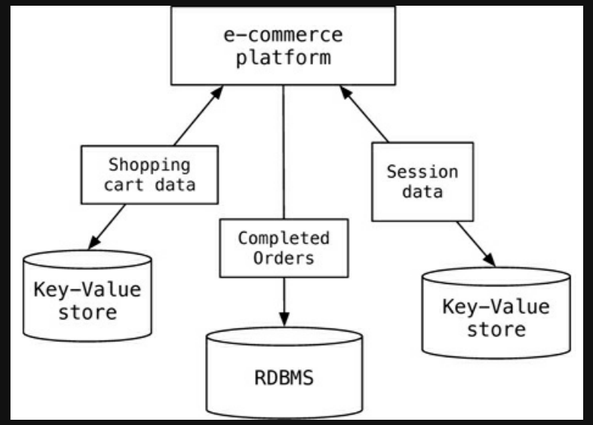


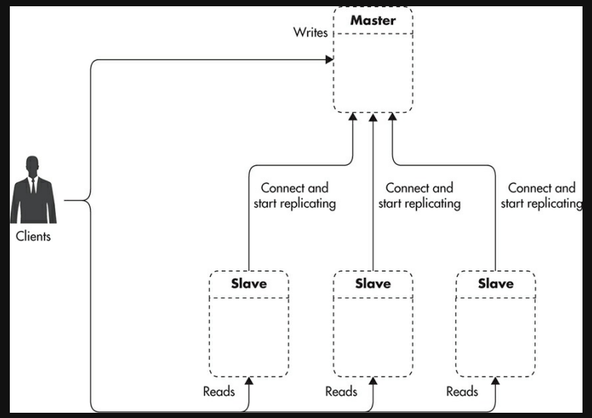
Architecture & Design ...

Microsoft Patterns & Practices
Microsoft Application & Architecture Guide
Shaping Software - Architecture
Clean MicroService Architecture
Coding the Architecture
Software & Systems Architecture

Architecture & Design ...

Choose Polyglot DataStores for different data needs



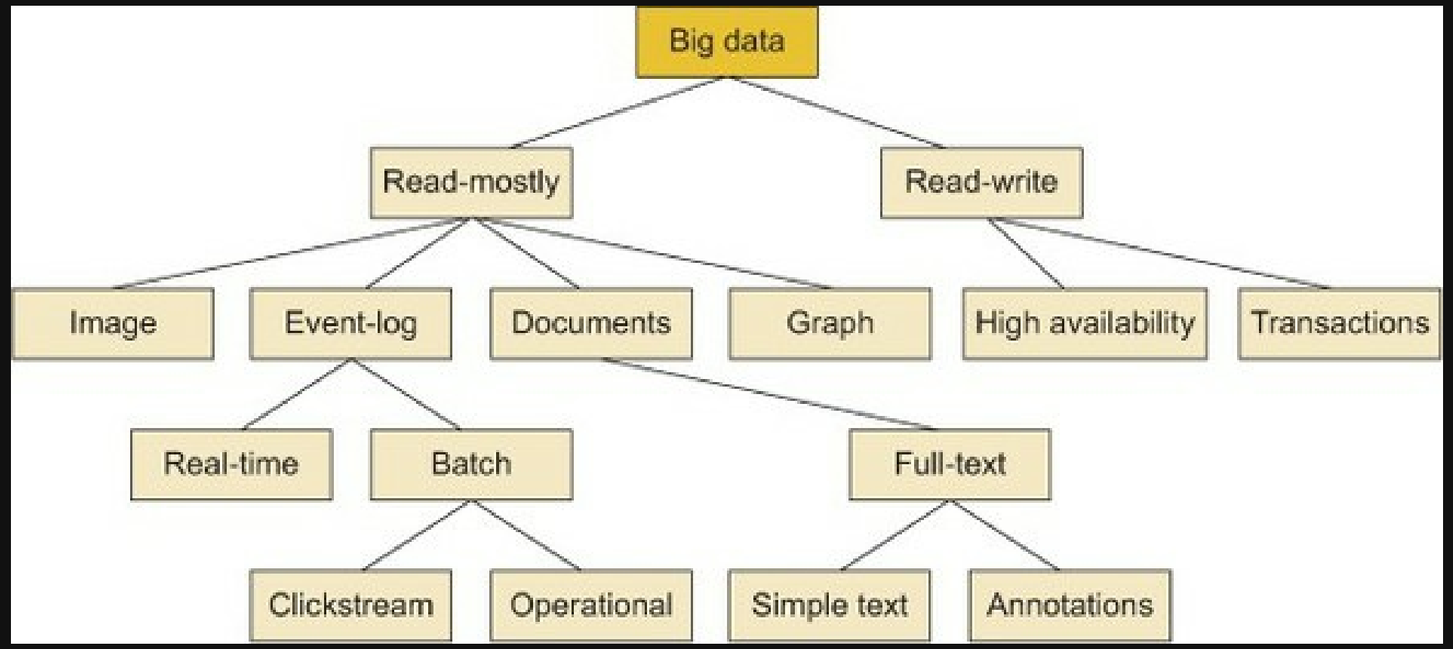


- RDBMS's for transactional needs (w replication), NoSQL for Scalability -

Microsoft Patterns & Practices - Data Guidance

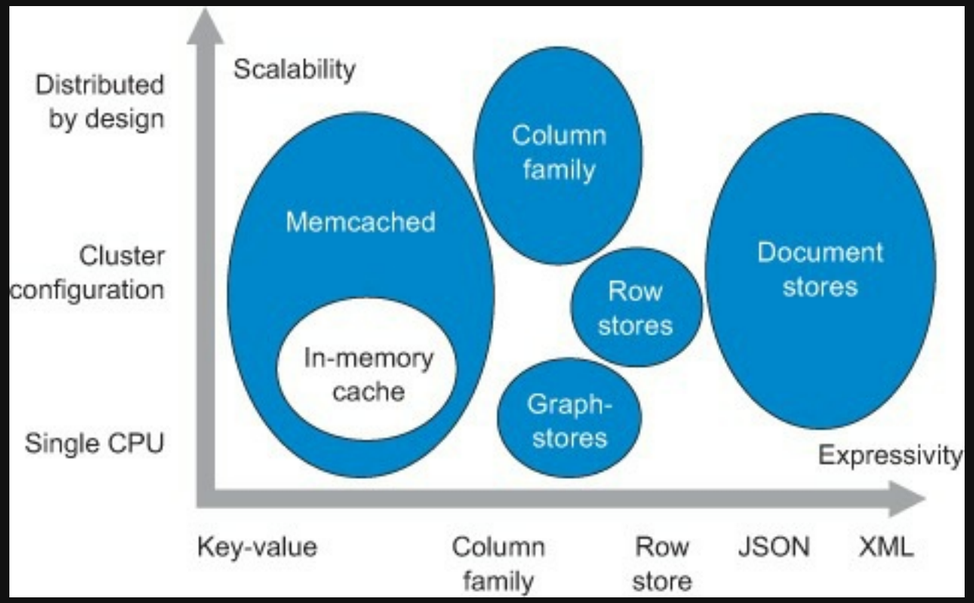
Architecture & Design ...

Tips to choose the right data store based on the use case

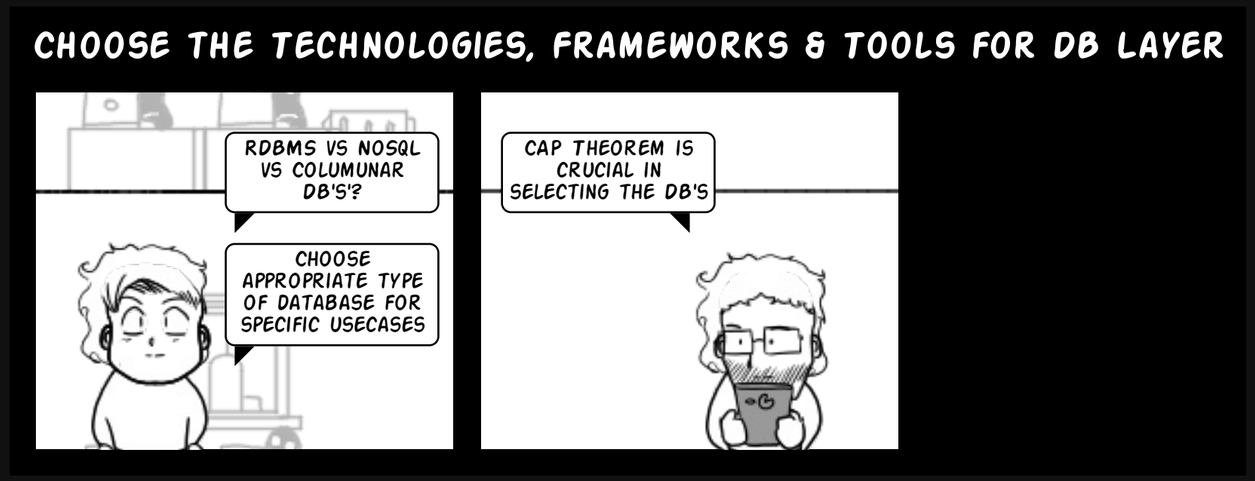


Architecture & Design ...

Tips to choose the right data store based on the type of data stored

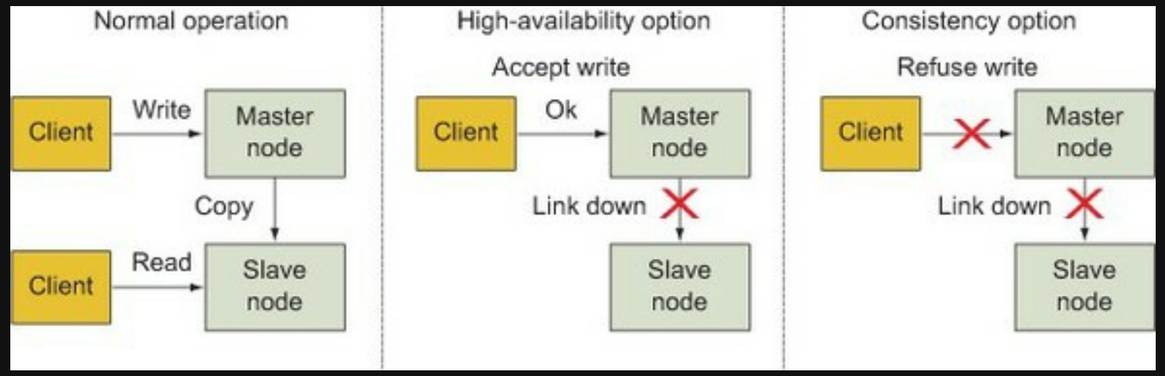


Architecture & Design ...



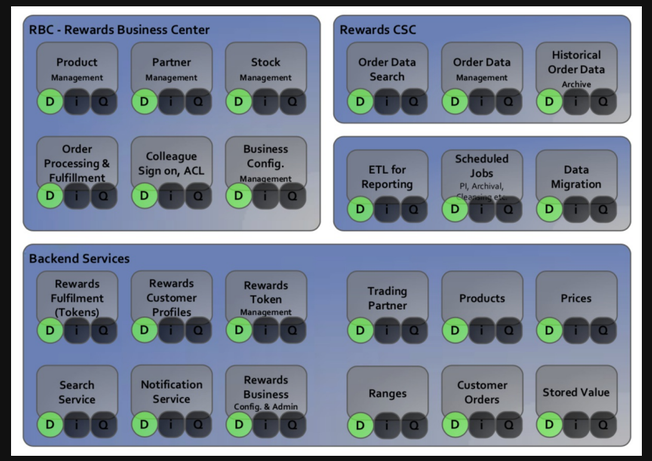
Architecture & Design ...

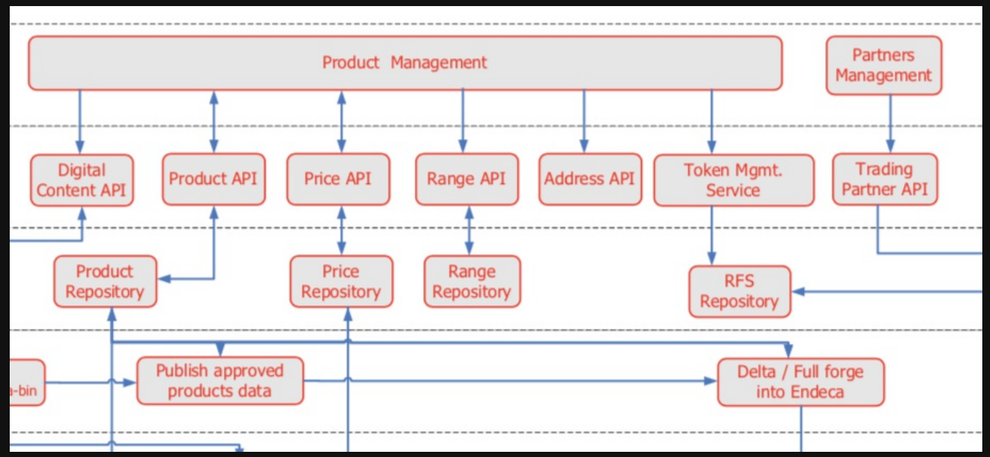
CAP Theorem



- Choosing between Availability, Consistency & Partition Tolerance -

Architecture & Design ...

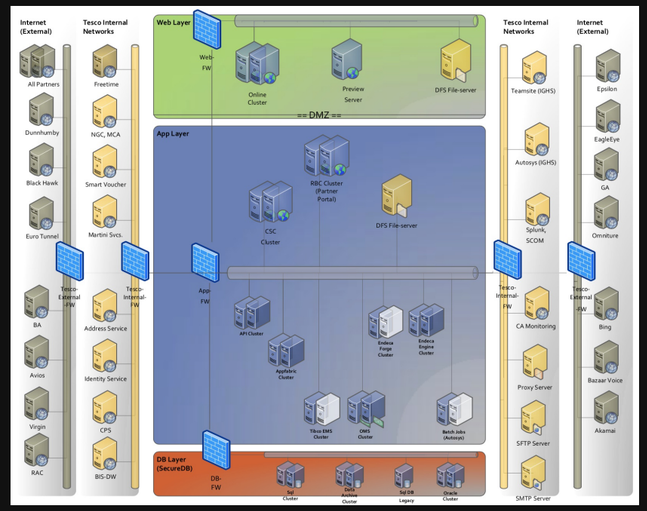




- Logical, Logical Connectivity Views -

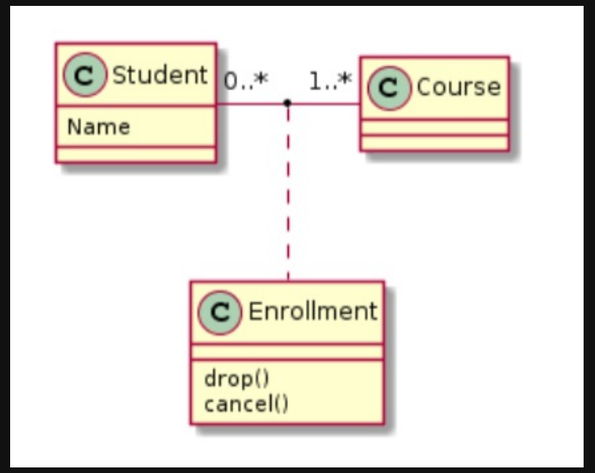
yED from yWorks

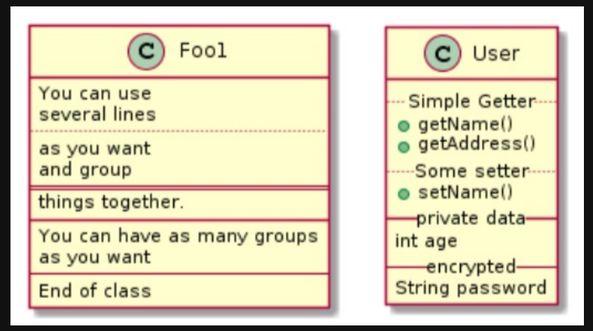
Architecture & Design ...

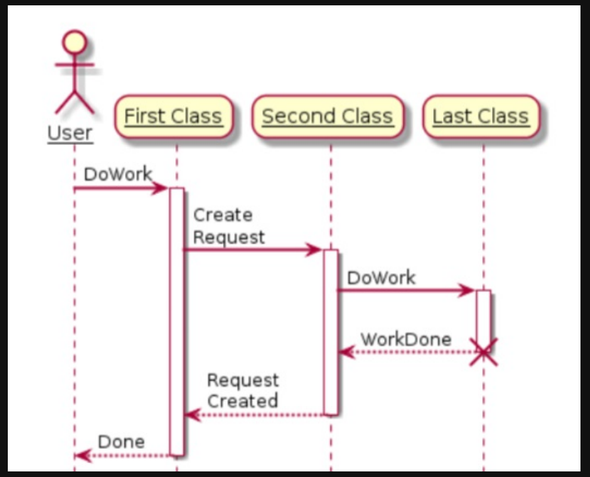


- Physical View -

Architecture & Design ...







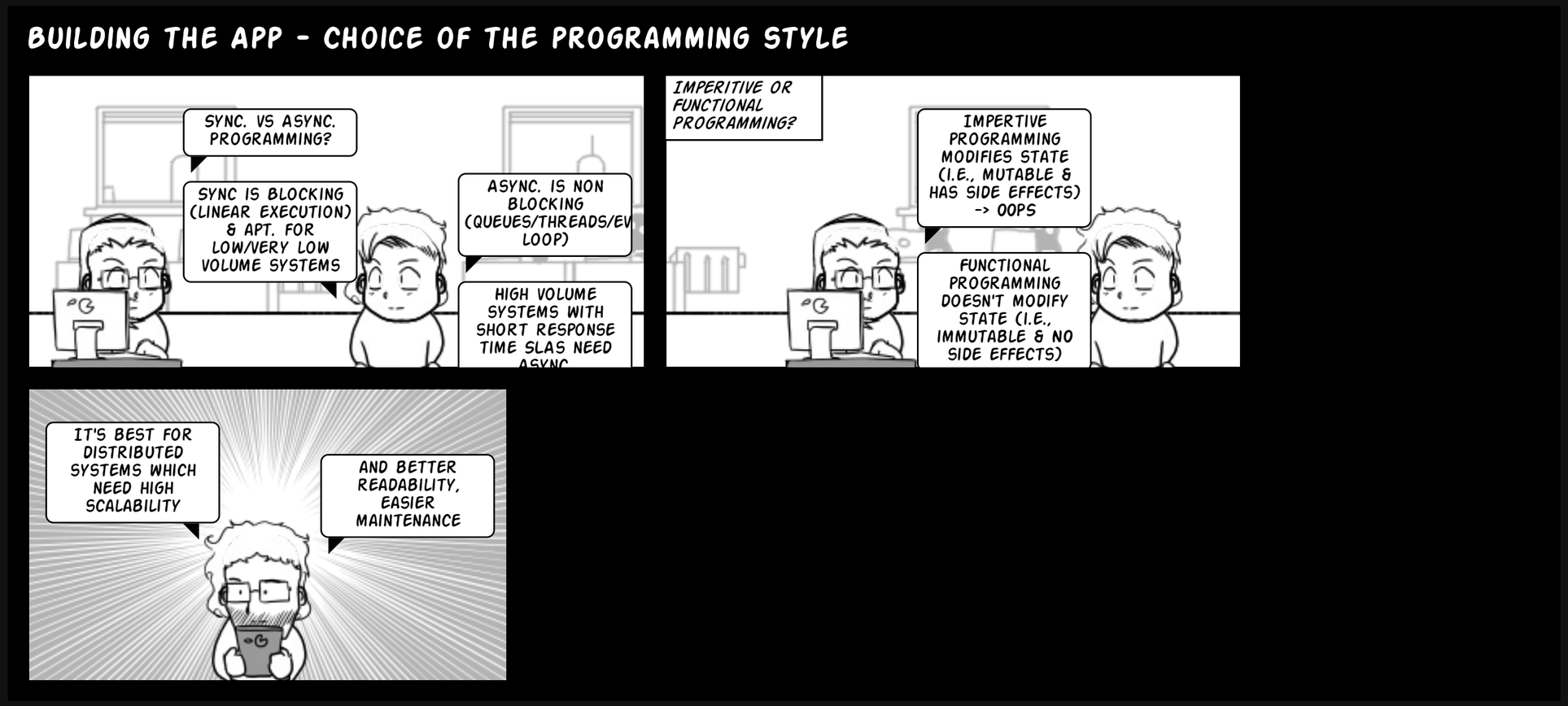
- Class & Sequence Diagrams -

PlantUML for Declarative diagrams
UML Simplified
ArgoUML
UMLet

Engineering ...

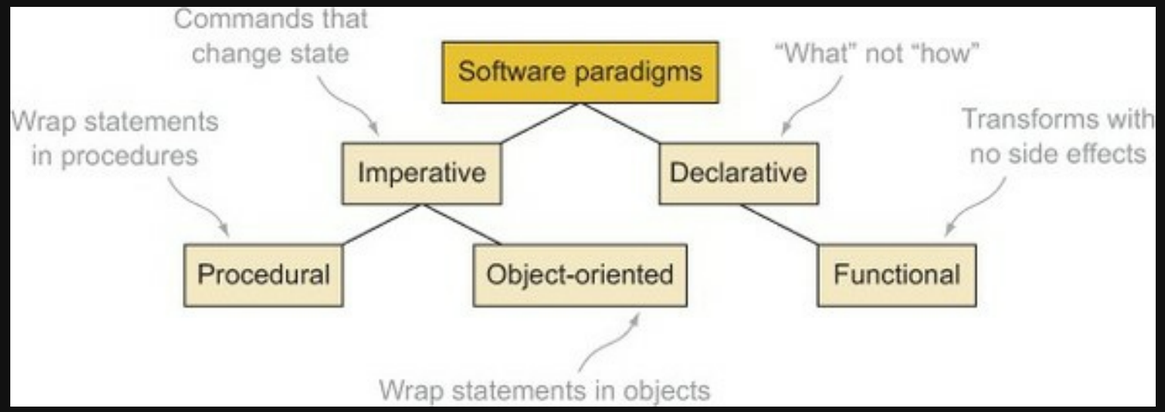
Follow Alistair Cockburn - Agile & Engineering Guru!
Microsoft Engineering Best Practices & Guidance

Engineering ...



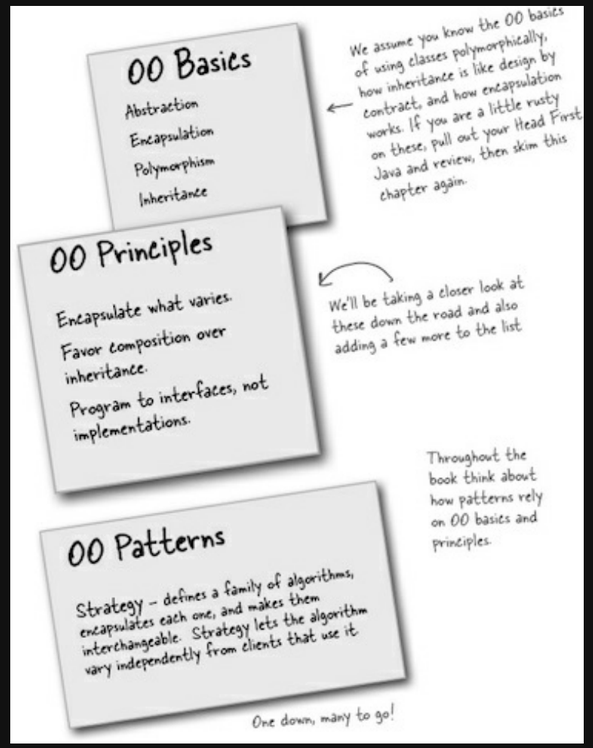
Engineering ...

Software Paradigms



Engineering ...

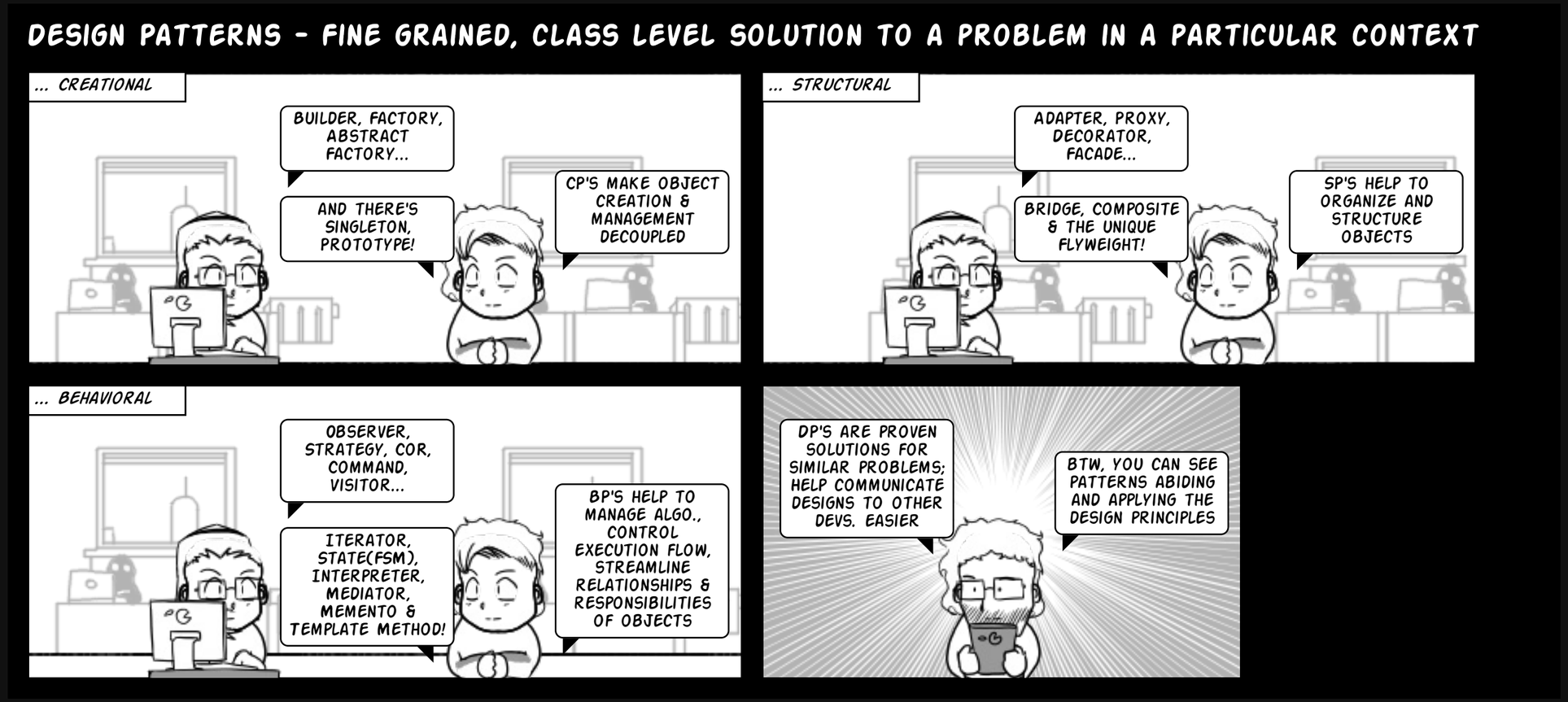
OOPS!



Engineering ...

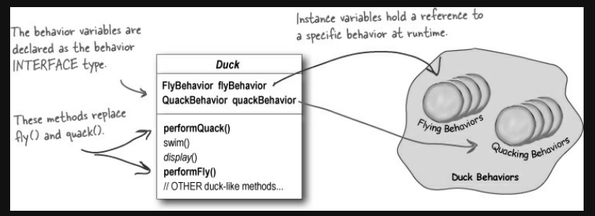
A Practical Intro to Functional Programming

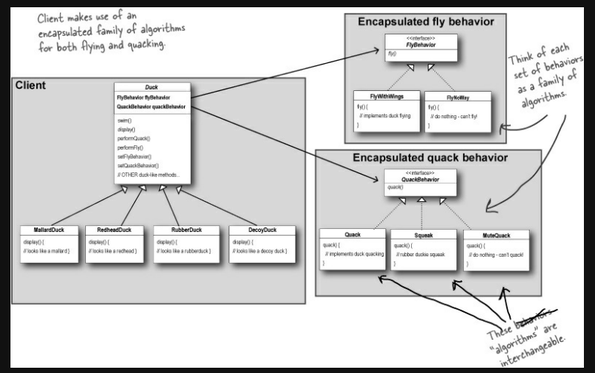
Engineering ...

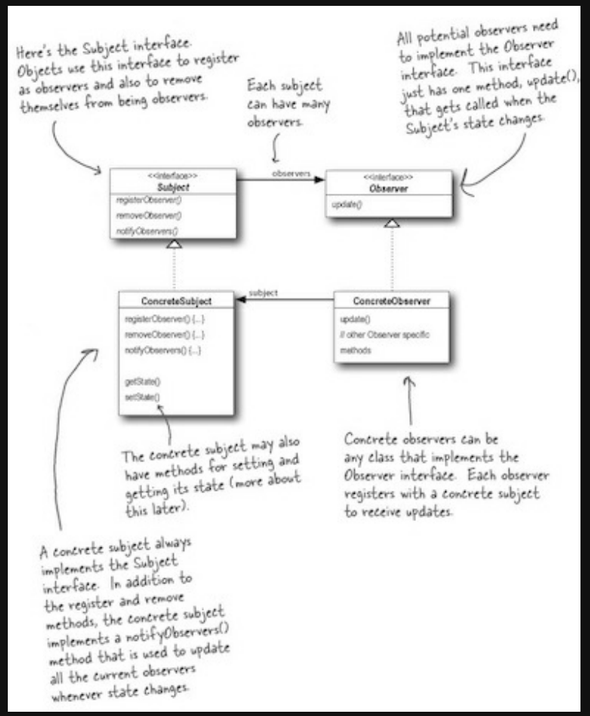


Design Patterns

Engineering ...

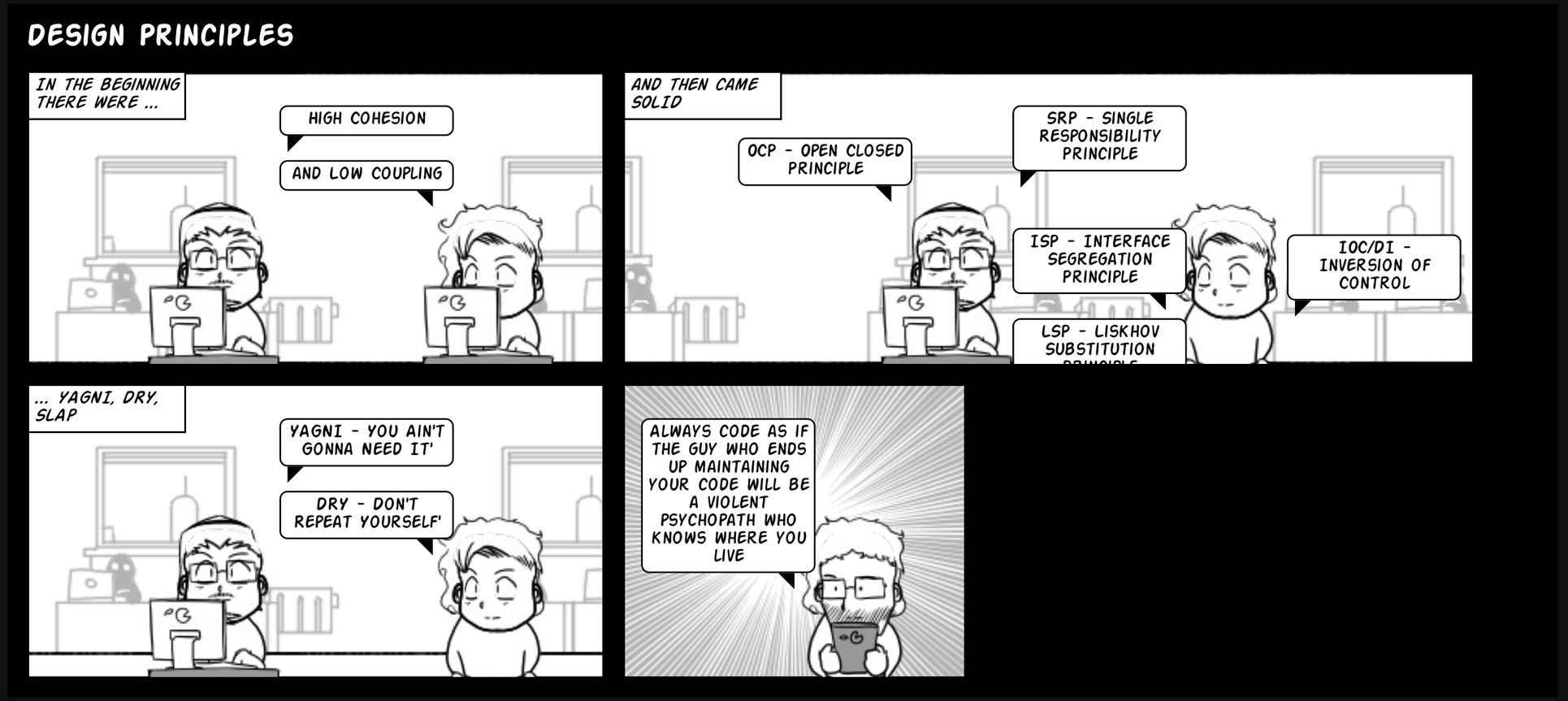






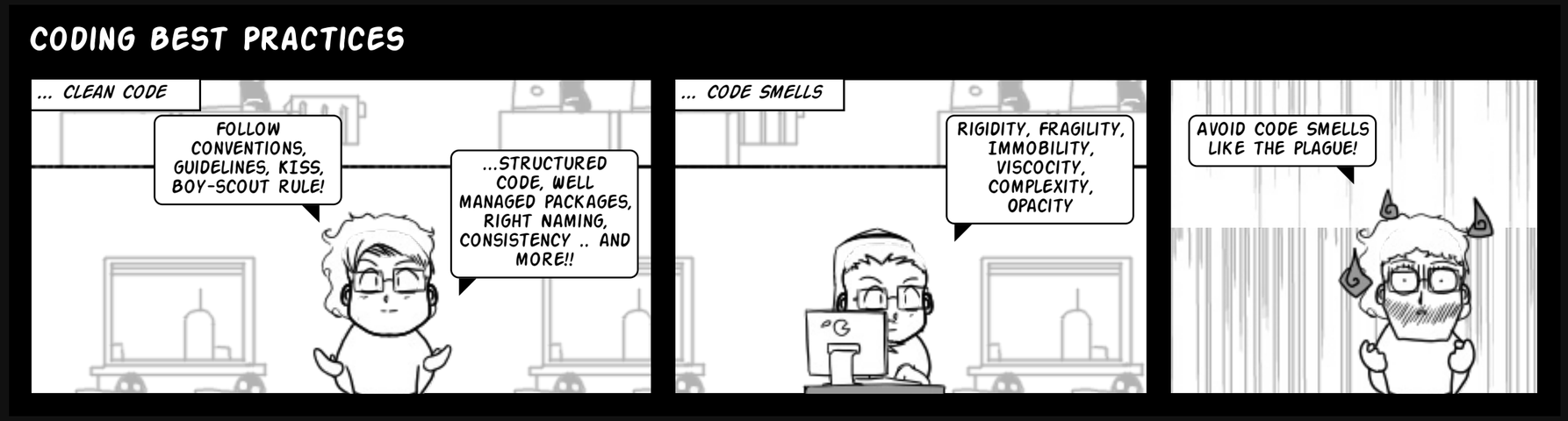
Design Patterns Catalog
Patterns & Principles w Examples - Exhaustive List

Engineering ...



Principles Wiki

NFR's : Quality....

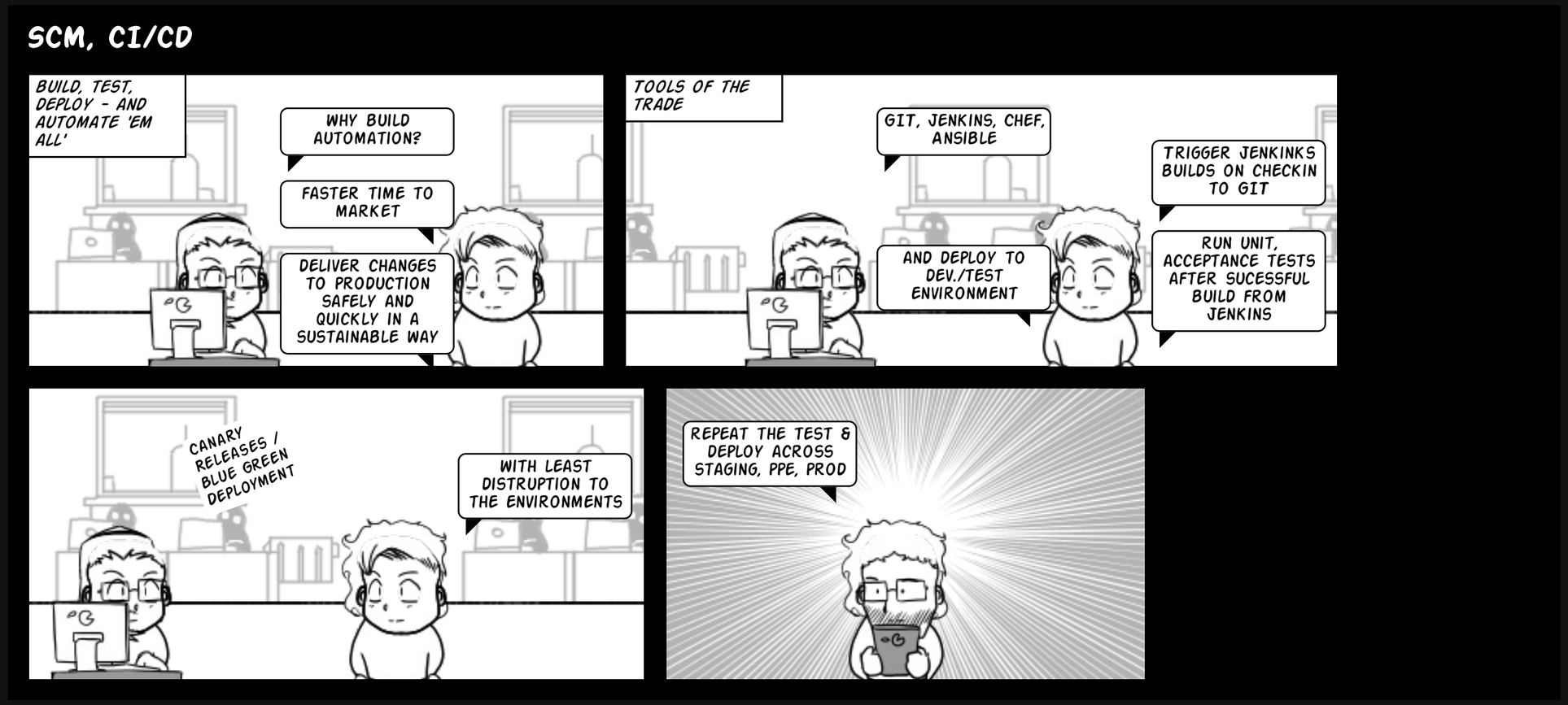


NFR's : Quality....

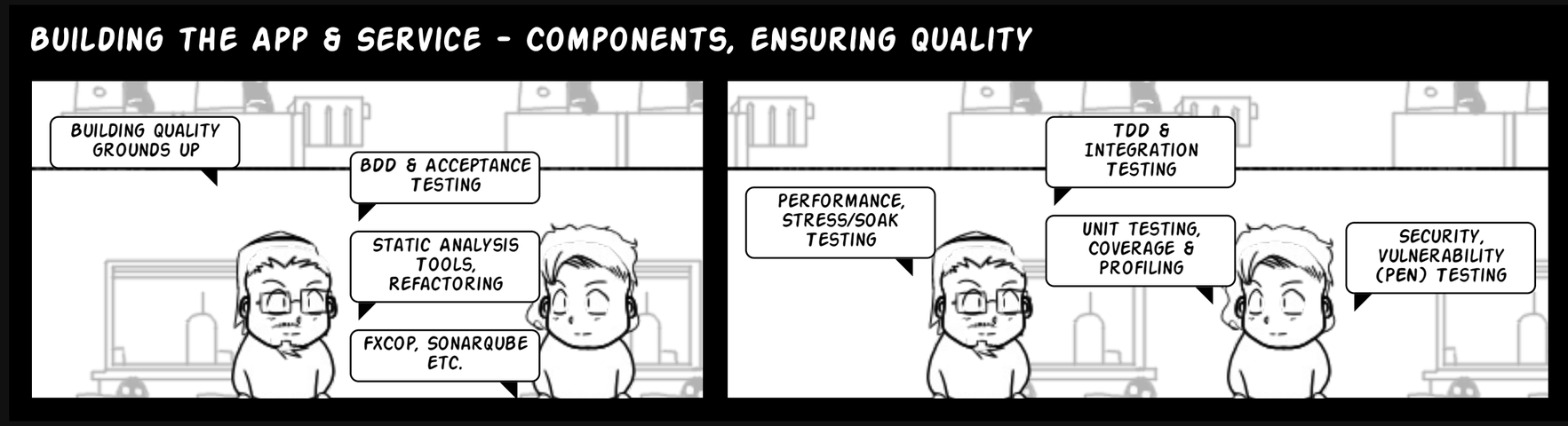
Clean Code Cheatsheet
Software Craftsmanship
Essential Programming Books - Bibles
Influential CS Books
Code Smells

NFR's : Quality....

NFR's : Quality....

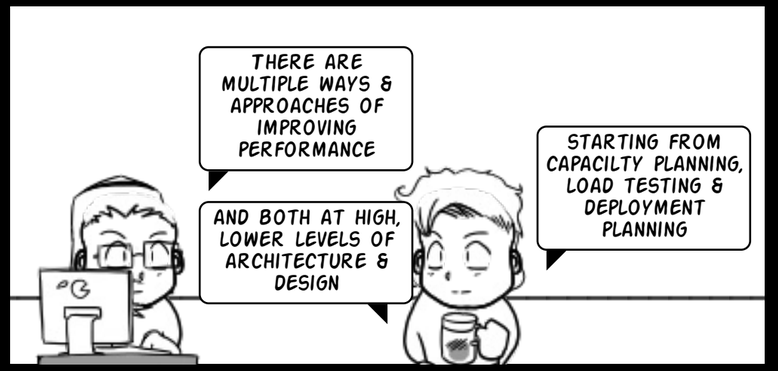


NFR's : Quality....



NFR's : Performance

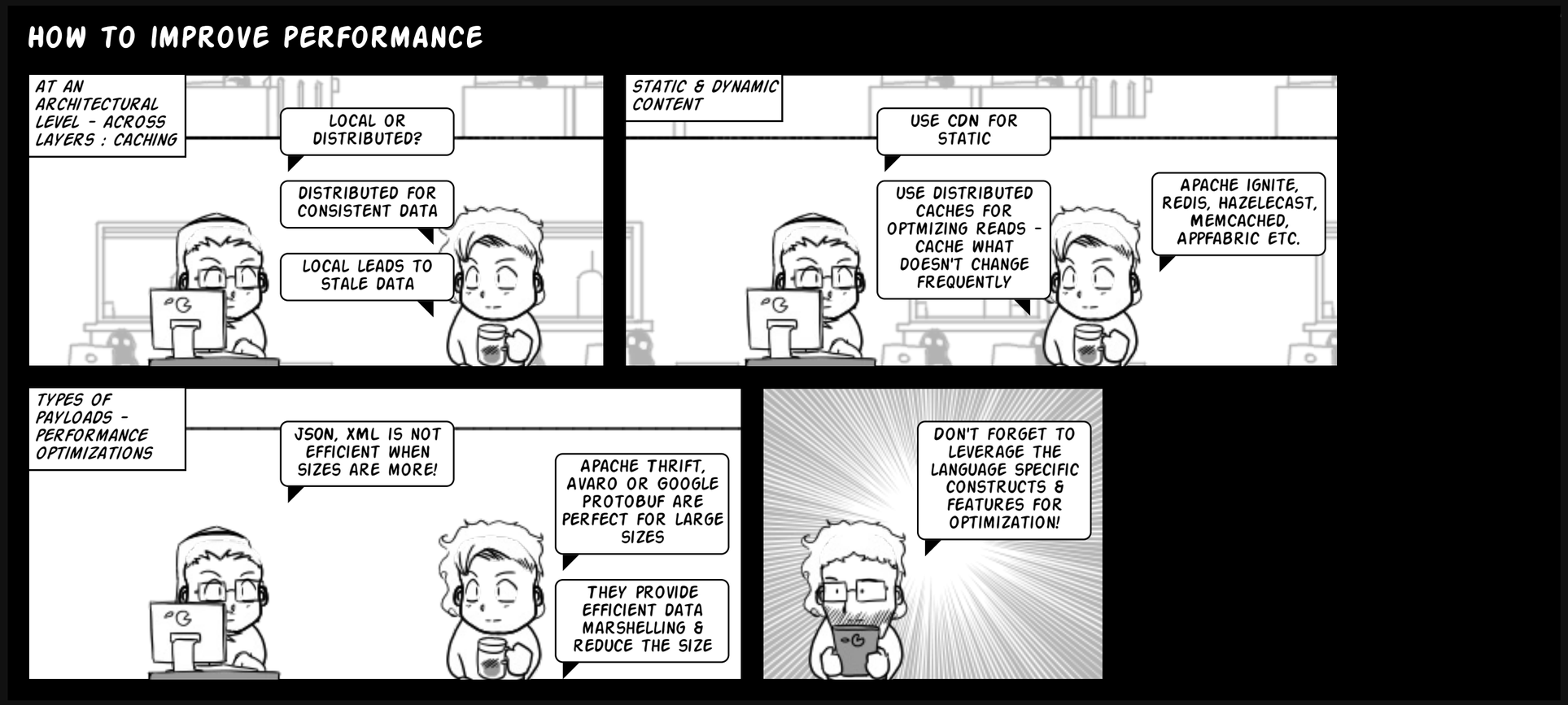
WHY IMPROVE PERFORMANCE?



NFR's : Performance

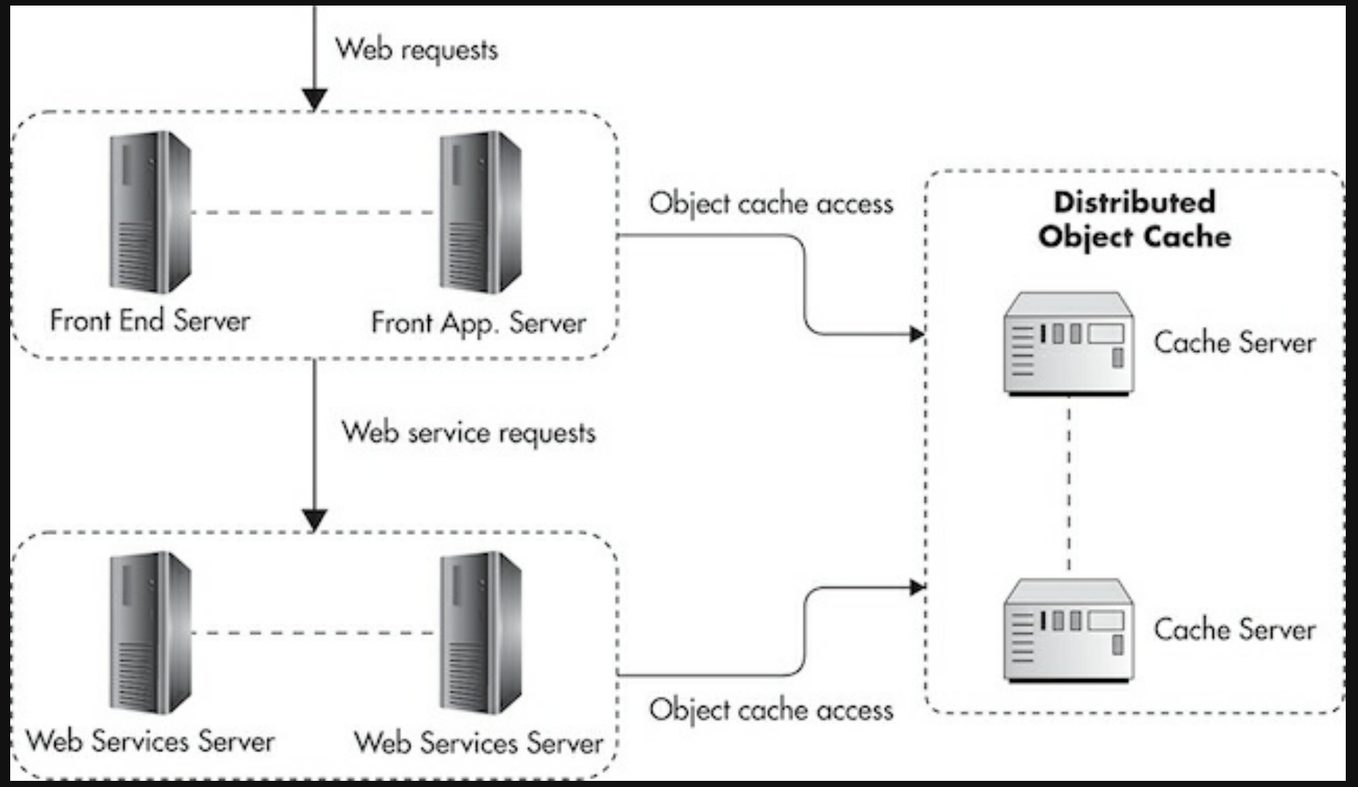
Performance Testing
Awesome Web Performance Optimization
Path To Performance - Podcast

NFR's : Performance

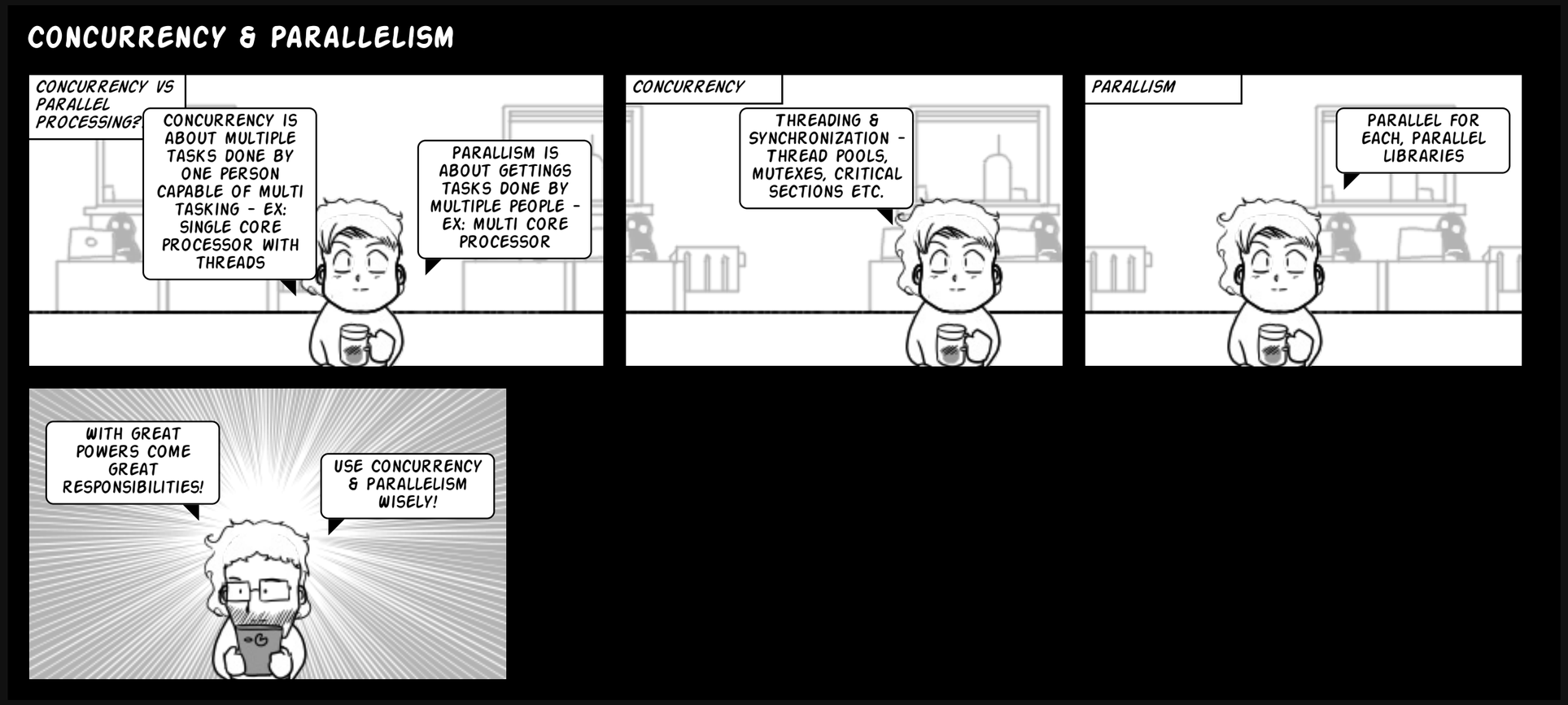


C# Internals
C# async/await
CLR via C#
Java Performance Tuning

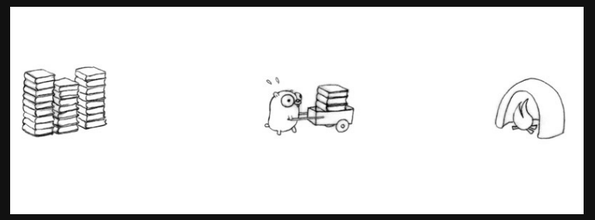
NFR's : Performance

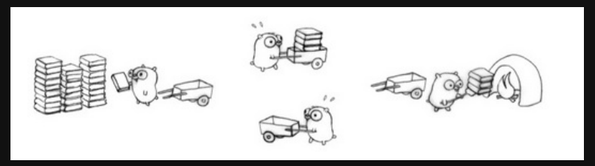


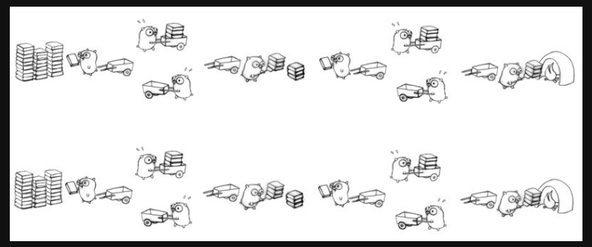
NFR's : Performance



NFR's : Performance

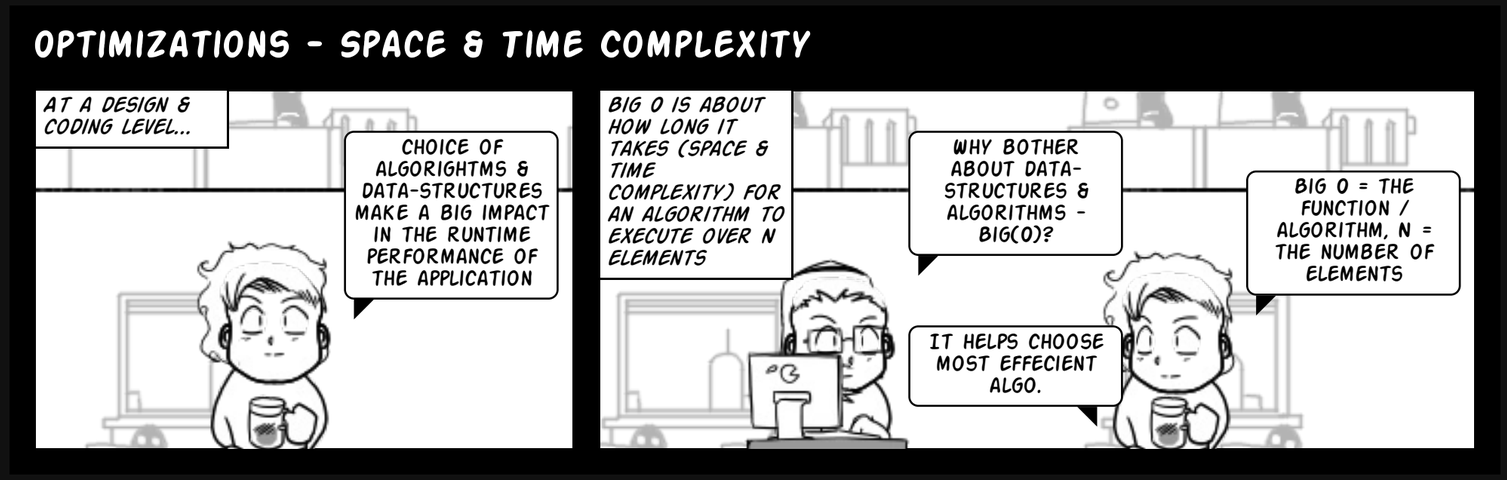




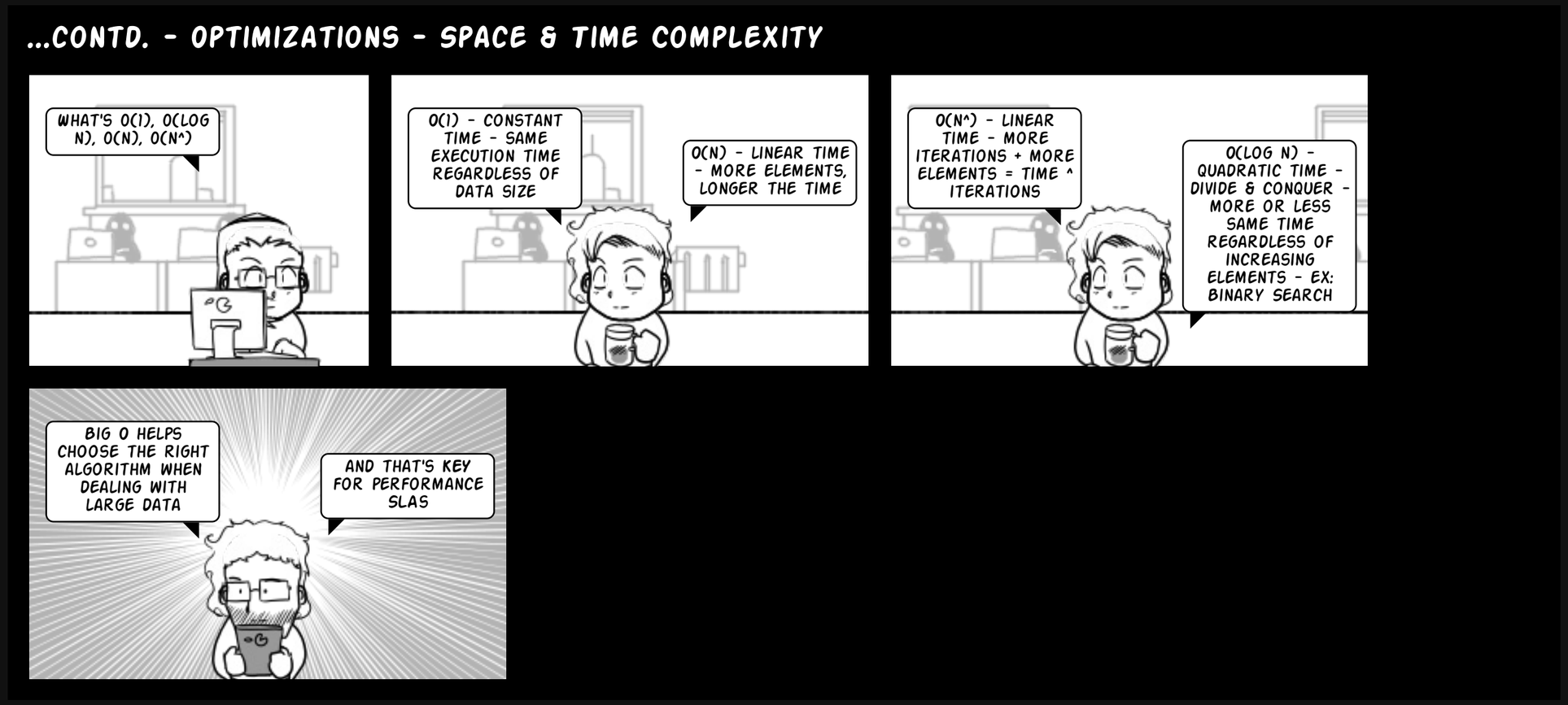


Concurrency vs Parallelism
Threading in C#

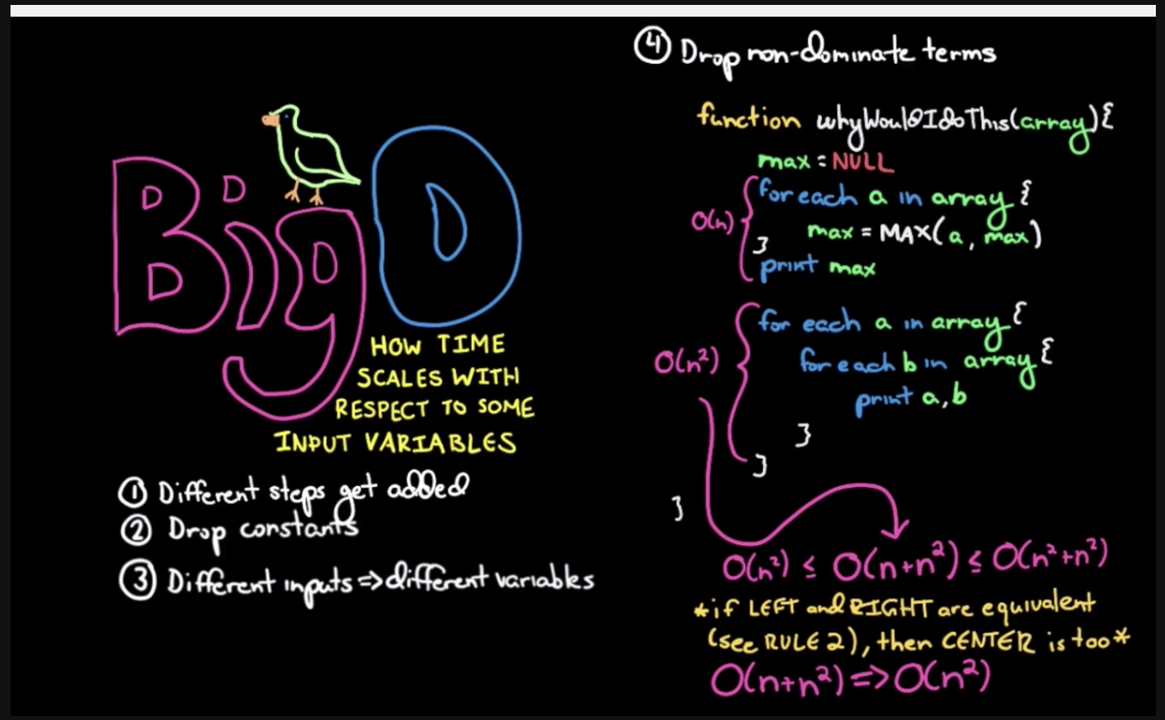
NFR's : Performance



NFR's : Performance



NFR's : Performance

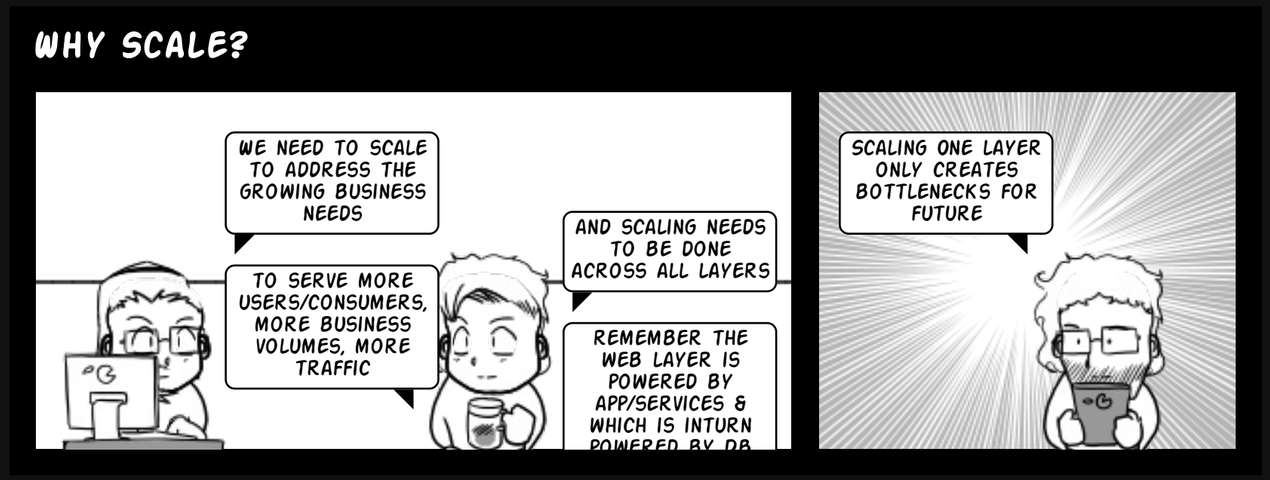


Grokking Algorithms!

NFR's : Performance

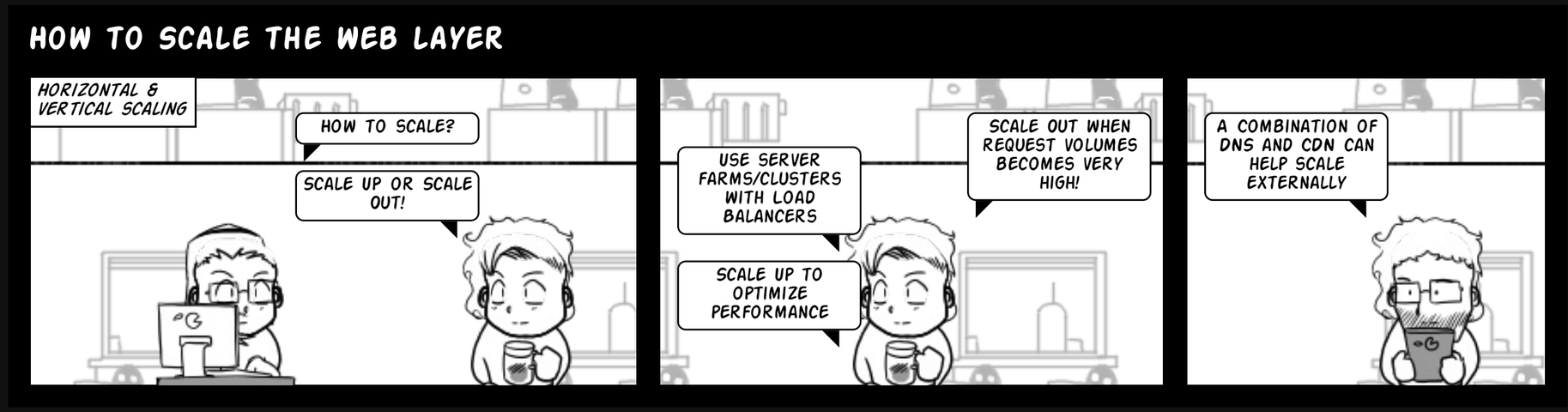
A Beginners guide to big-o-notation
Big-O notation in 5 minutes — The basics
Big O Notation explained
Big O Notation: A Few Examples
Big O CheatSheet
Algorithms for dummies
Algorithimic Complexity
Understanding Big O

NFR's : Scalability

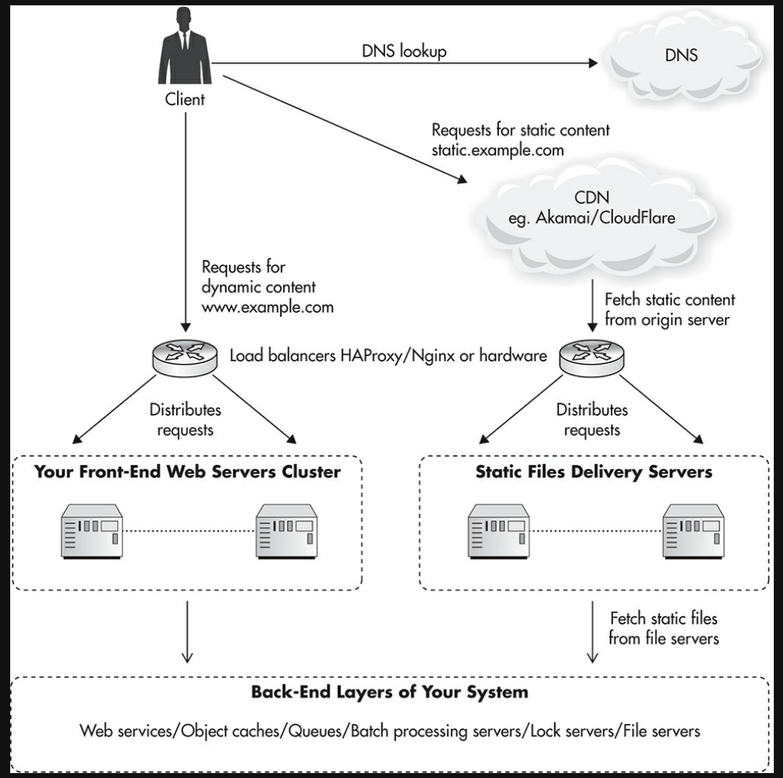


Awesome Scalability
Microsoft Improving Perfmormance & Scalability

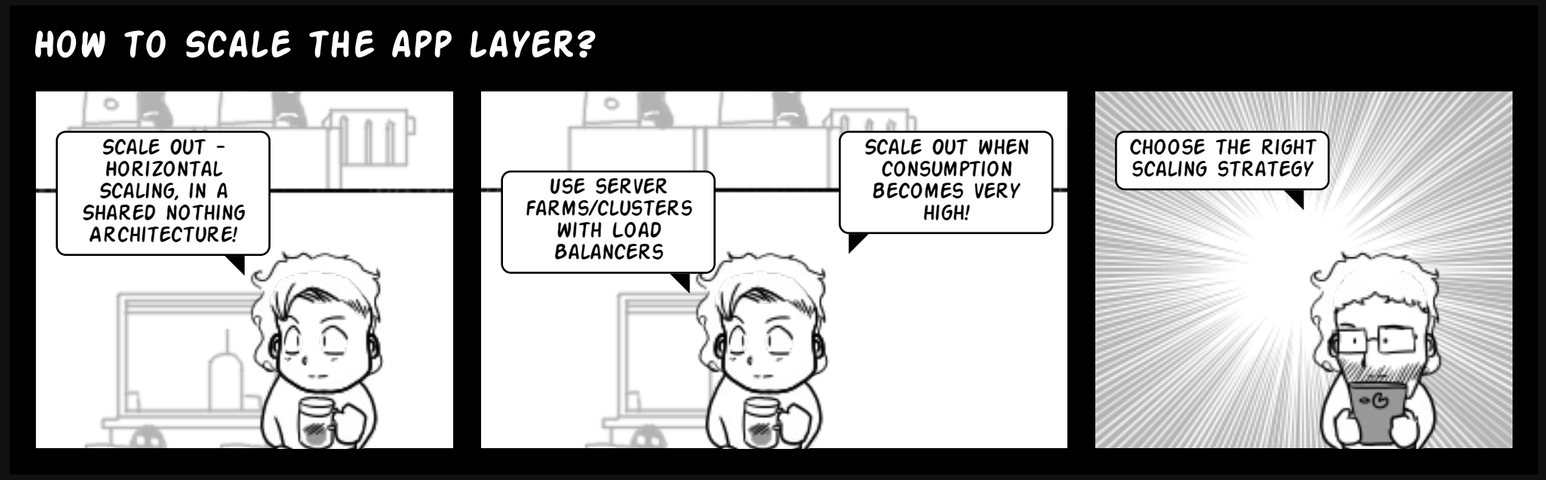
NFR's : Scalability



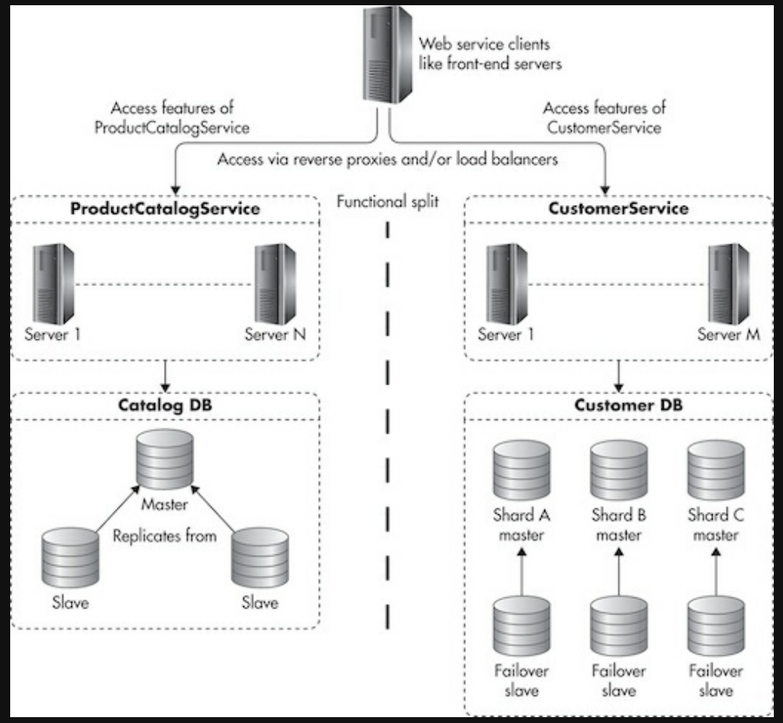
NFR's : Scalability



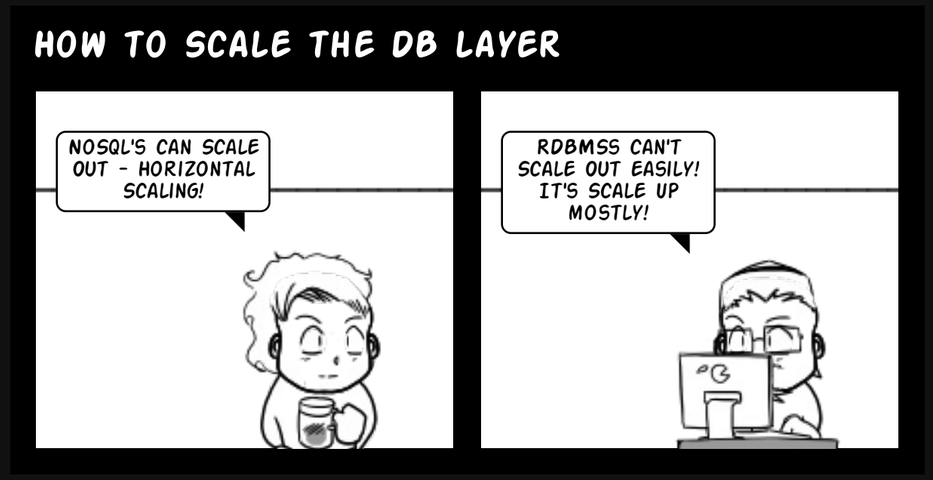
NFR's : Scalability



NFR's : Scalability

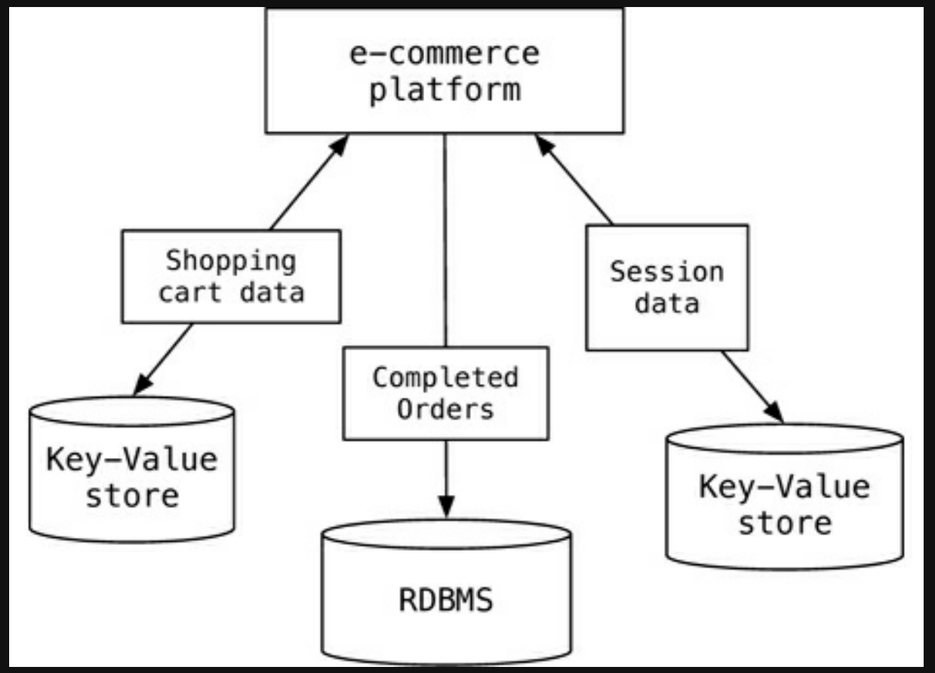


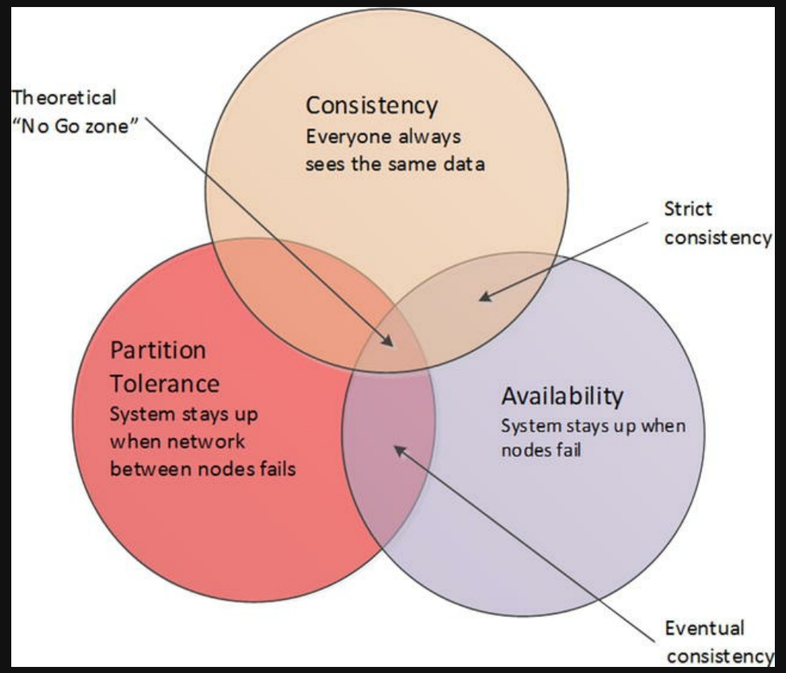
NFR's : Scalability



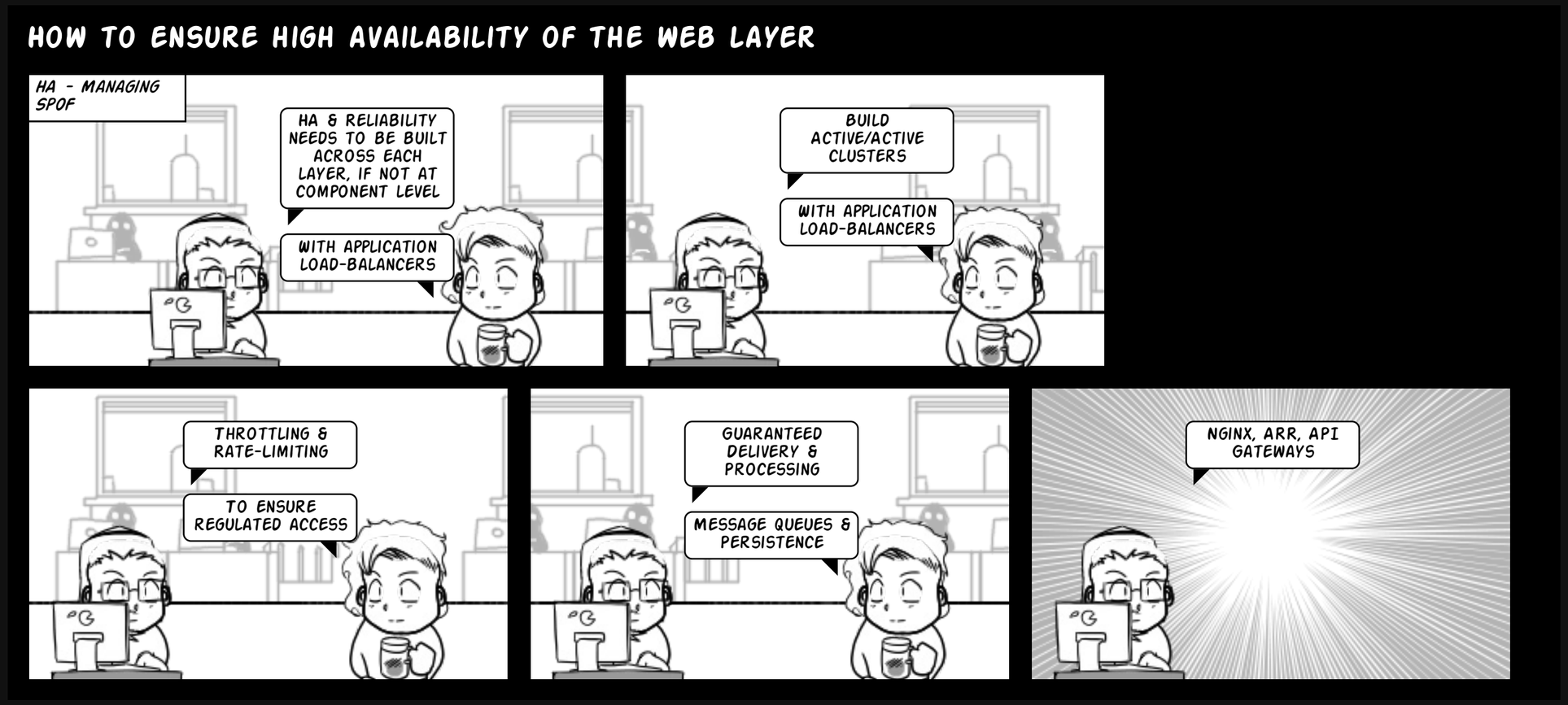
NFR's : Scalability

- Revisiting DB Scaling + CAP -





NFR's : HA & Reliability

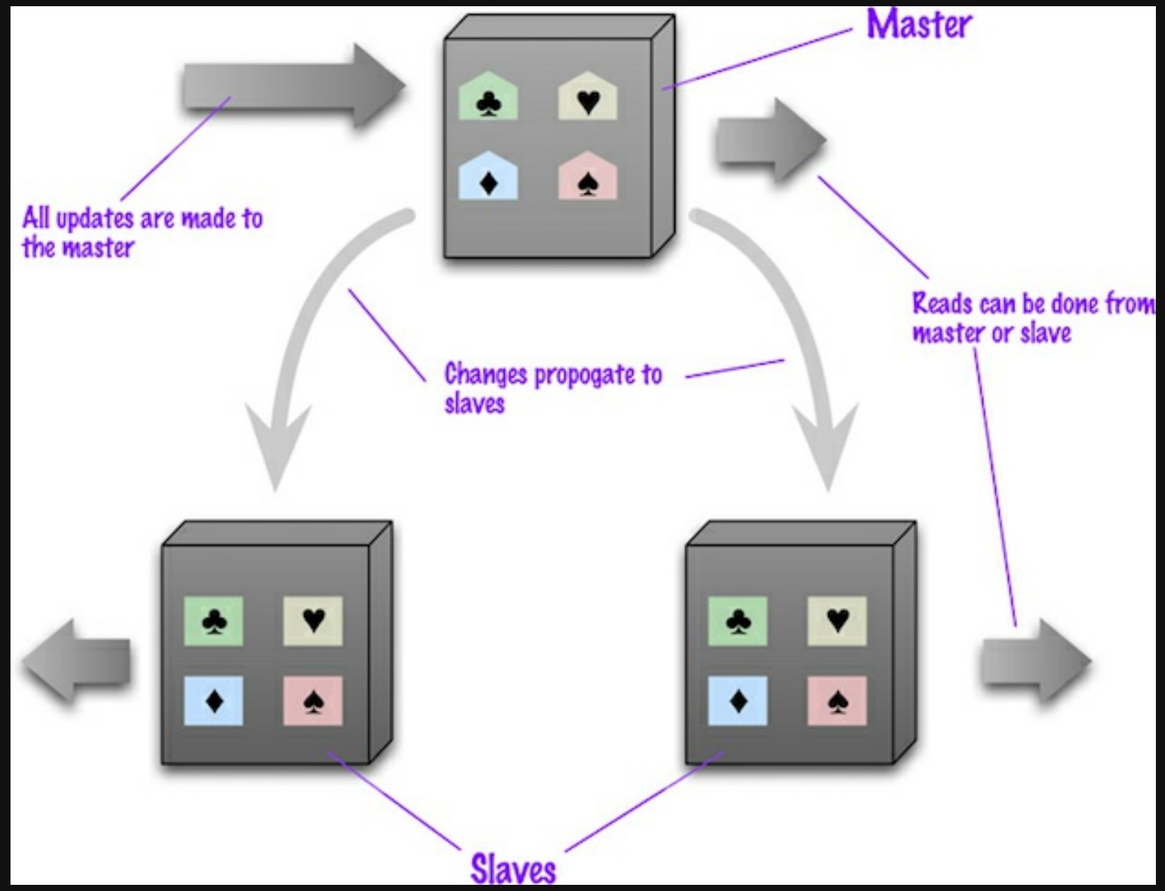


NFR's : HA & Reliability

NFR's : HA & Reliability

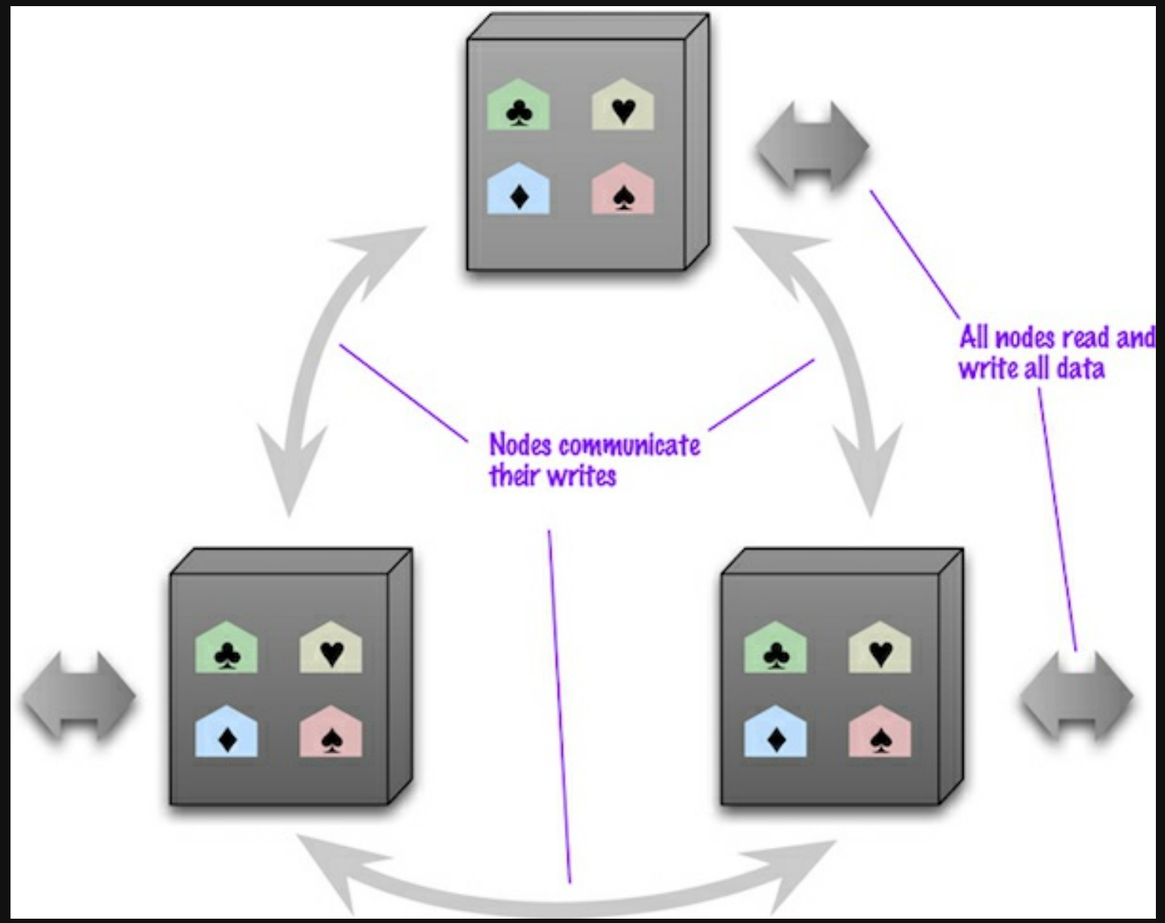
Scalability + HA go hand in hand!
HA Calls for more than one server;
An Active / Active HA cannot be possible without either Scale UP or Scale Out

NFR's : HA & Reliability



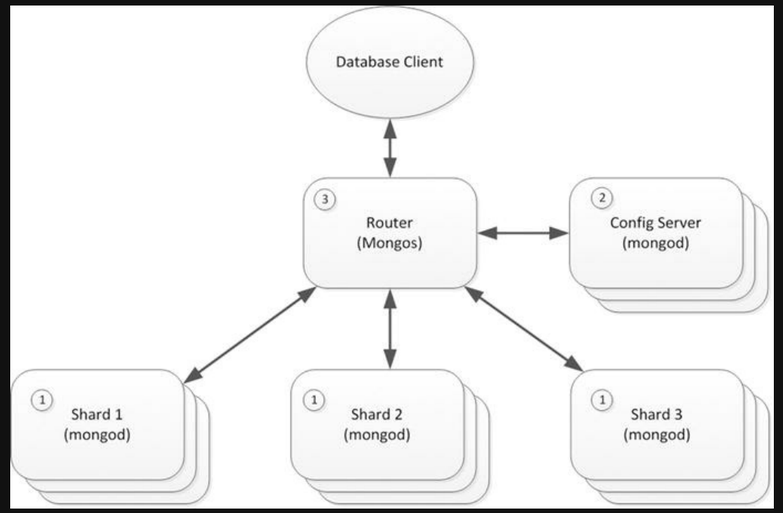
Master Slave (Active / Passive)

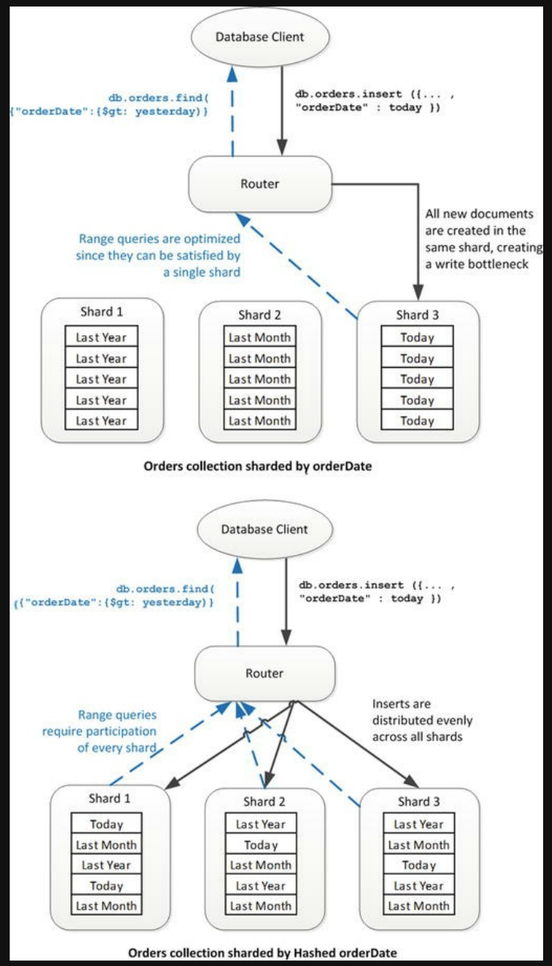
NFR's : HA & Reliability



Master Slave : Peer - Peer (Active / Active)

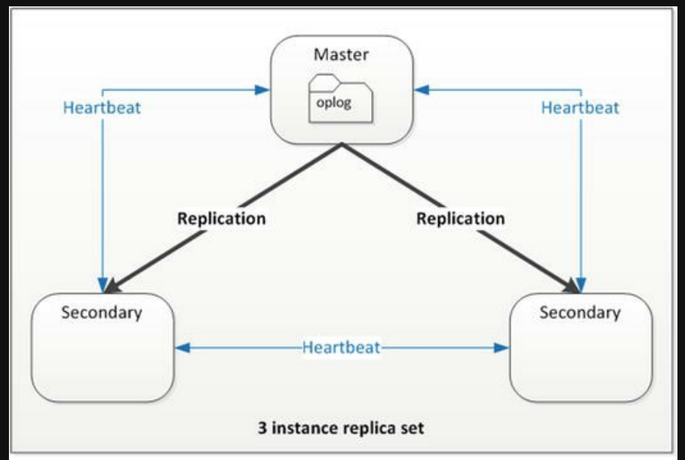
NFR's : HA & Reliability

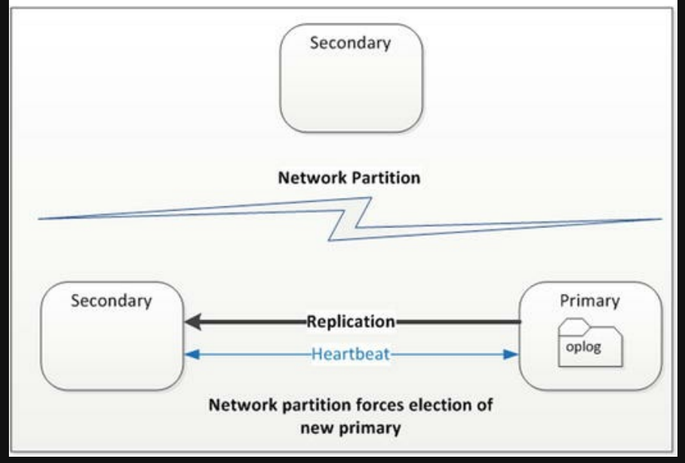




Sharding a NoSQL Database (Partition Data to manage scale)

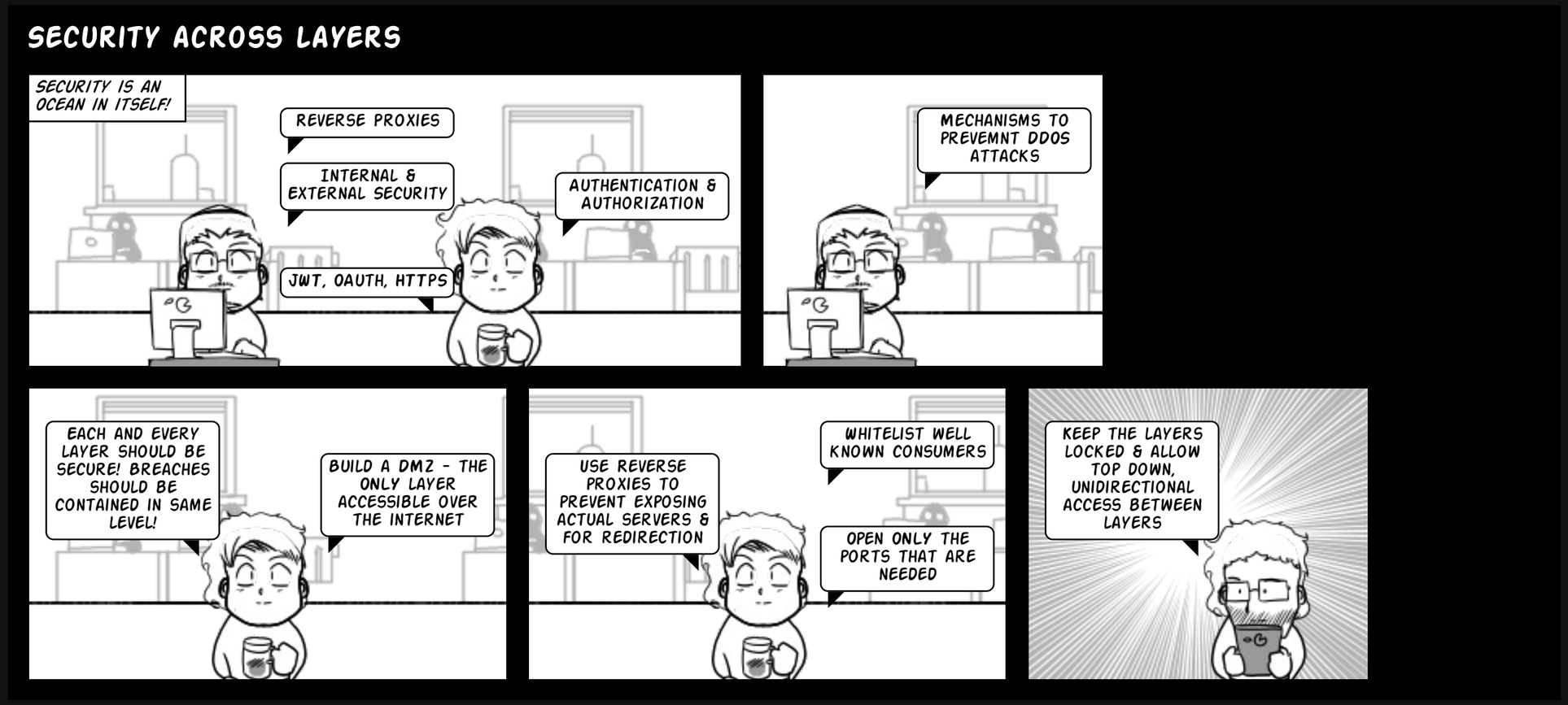
NFR's : HA & Reliability



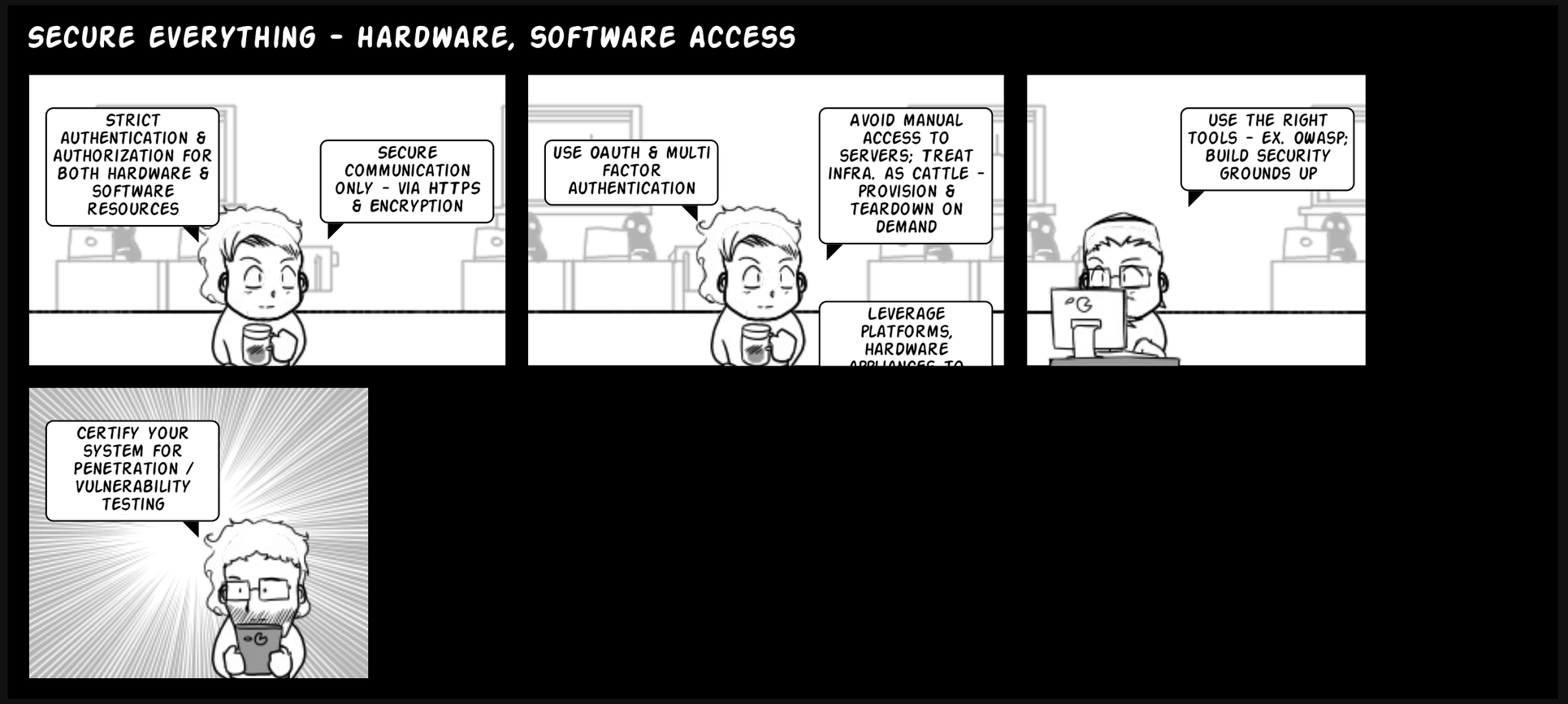


Replication set (Active / Active, Active / Passive)

NFR's : Securing the System



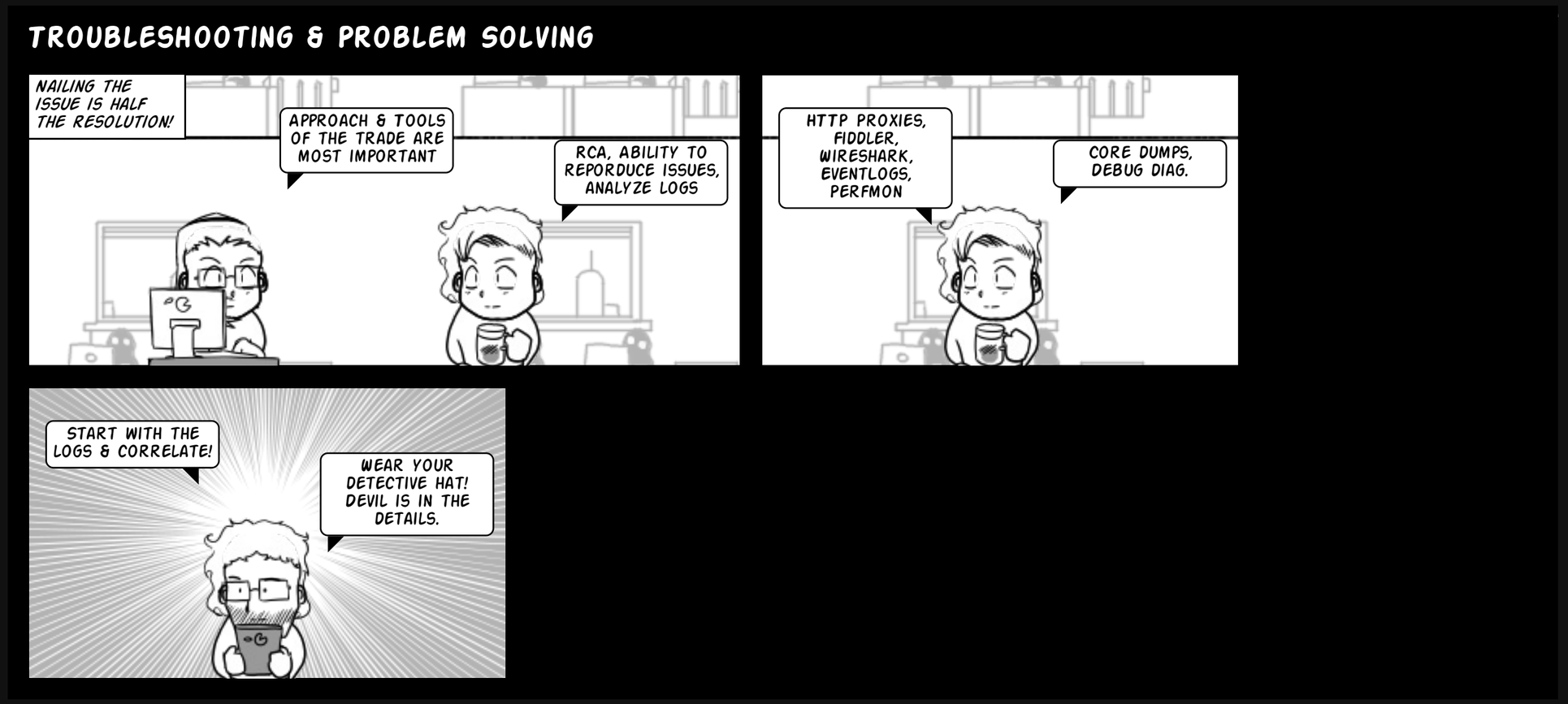
NFR's : Securing the System



NFR's : Securing the System

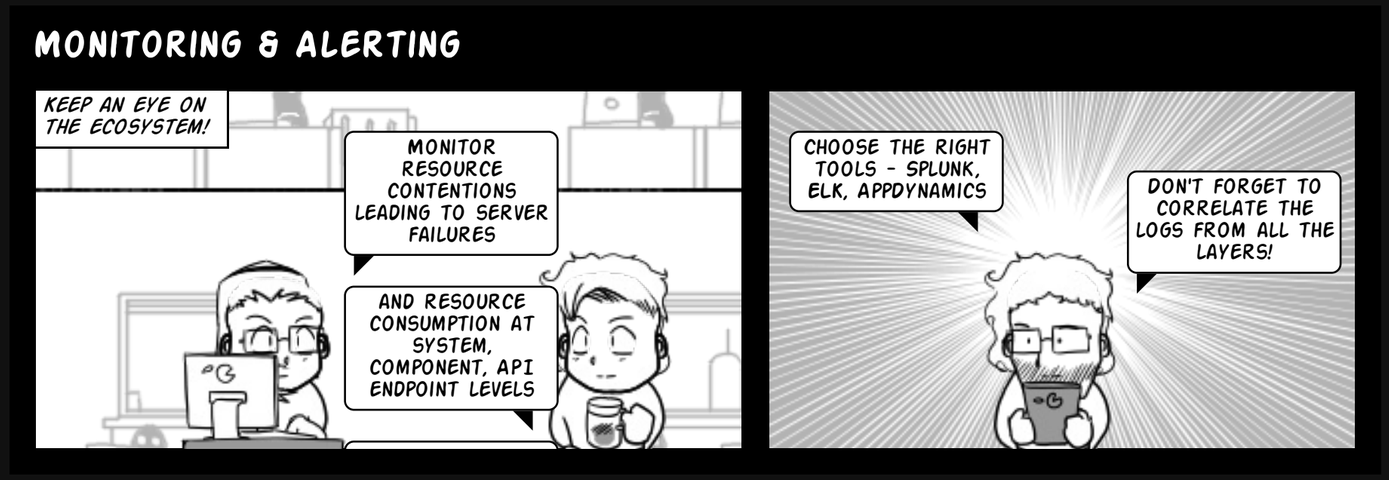
Microsoft Patterns & Practices Guidance - Security
Microsoft Web Application Security: Threats and Countermeasures
Azure Security
OWASP
Vulnerability Testing vs Penetration Testing

Operational needs



JConsole
VisualVM - All in One Java troubleshooting tool
JRockit Mission Control
Debugging .Net Framework Apps

Operational needs



Links

Donne Martin's System Design Primer

- The Single Most absolute reference and
superbly organized collection of resources for system design -

Links

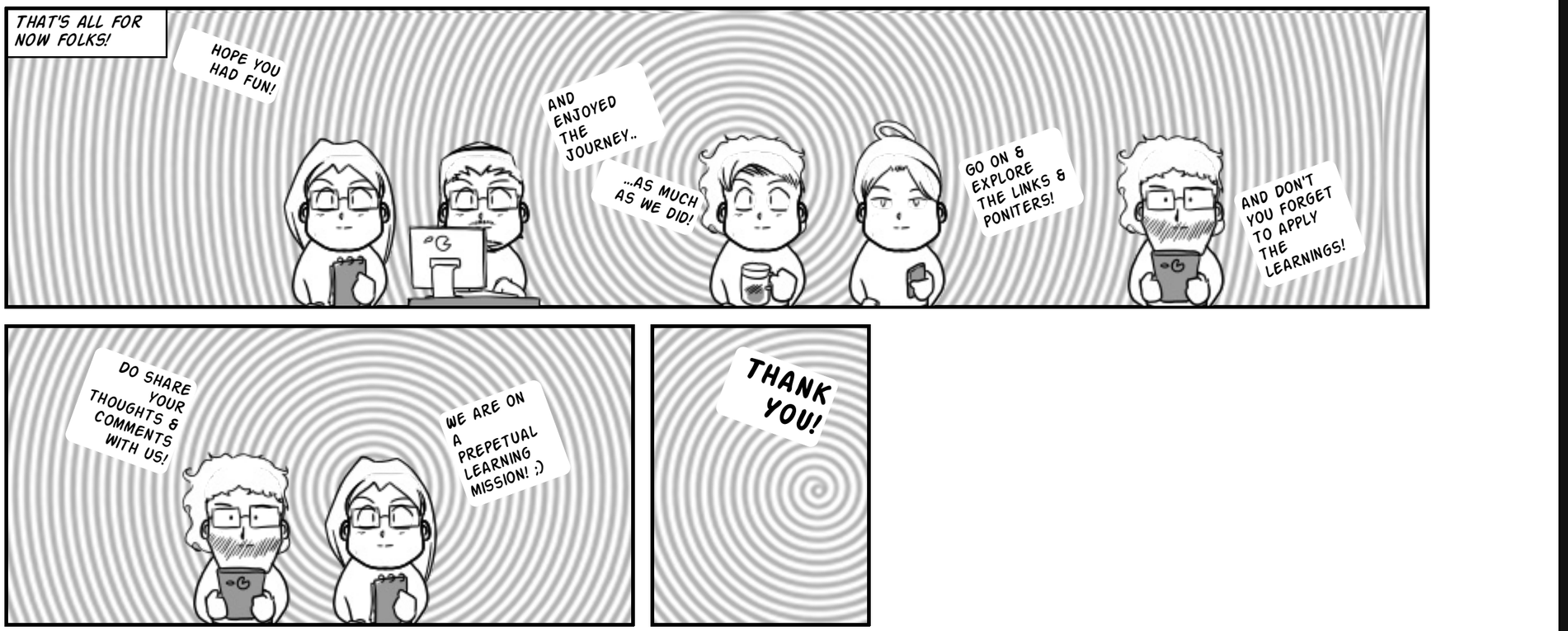
Sources every engineer & architect should follow!

InfoQ Tech. Hub
High Scalability - Real Time Architectures
DZone Tech. Hub
HackerNoon - Bleeding edge of tech.
Hacker News

Links

There's much much more, and these links give you good starting points

NetFlix OSS
Everything about MicroServices
Cloud Foundry
Cloud Native Computing Foundation
A curated list of awesome resources
Foundations of Software Engineering
System Design Inputs
Awesome Software Craftsmanship
Awesome BigData - All about data stores & databases
Microservice Patterns


THAT'S ALL FOR NOW FOLKS!
HOPE YOU HAD FUN!
AND ENJOYED THE JOURNEY..
...AS MUCH AS WE DID!
GO ON & EXPLORE THE LINKS & PONITERS!
AND DON'T YOU FORGET TO APPLY THE LEARNINGS!
DO SHARE YOUR THOUGHTS & COMMENTS WITH US!
WE ARE ON A PREPETUAL LEARNING MISSION! ;)
THANK YOU!

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

Visit Jim Hunt at **facebook.com/huntcartoons**