SOFTWARE ARCHITECTURE

DR. VIVEK NALLUR

VIVEK.NALLUR@SCSS.TCD.IE

https://www.scss.tcd.ie/vivek.nallur/teaching/slides/

ASSIGNMENT 1: CLARIFICATIONS

- There will be no trick inputs. Inputs will be valid for type and number. All inputs will be on standard input and be entered correctly.
- All field-size, simulation-time, bug-birth-time are zero-based
- Bugs will 'move' to their start position at their start time. In one time-instant, a bug will move and infect, in that order.

OUTLINE OF THIS TALK

- Software Architecture
  - Definition
  - Perspectives
  - Attributes

WHAT IS ARCHITECTURE?

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

ISO/IEC/IEEE 42010

WHAT IS ARCHITECTURE? -- OTHER DEFINITIONS

> *The software architecture of a program or computing system is a depiction of the system that aids in the understanding of how the system will behave.*

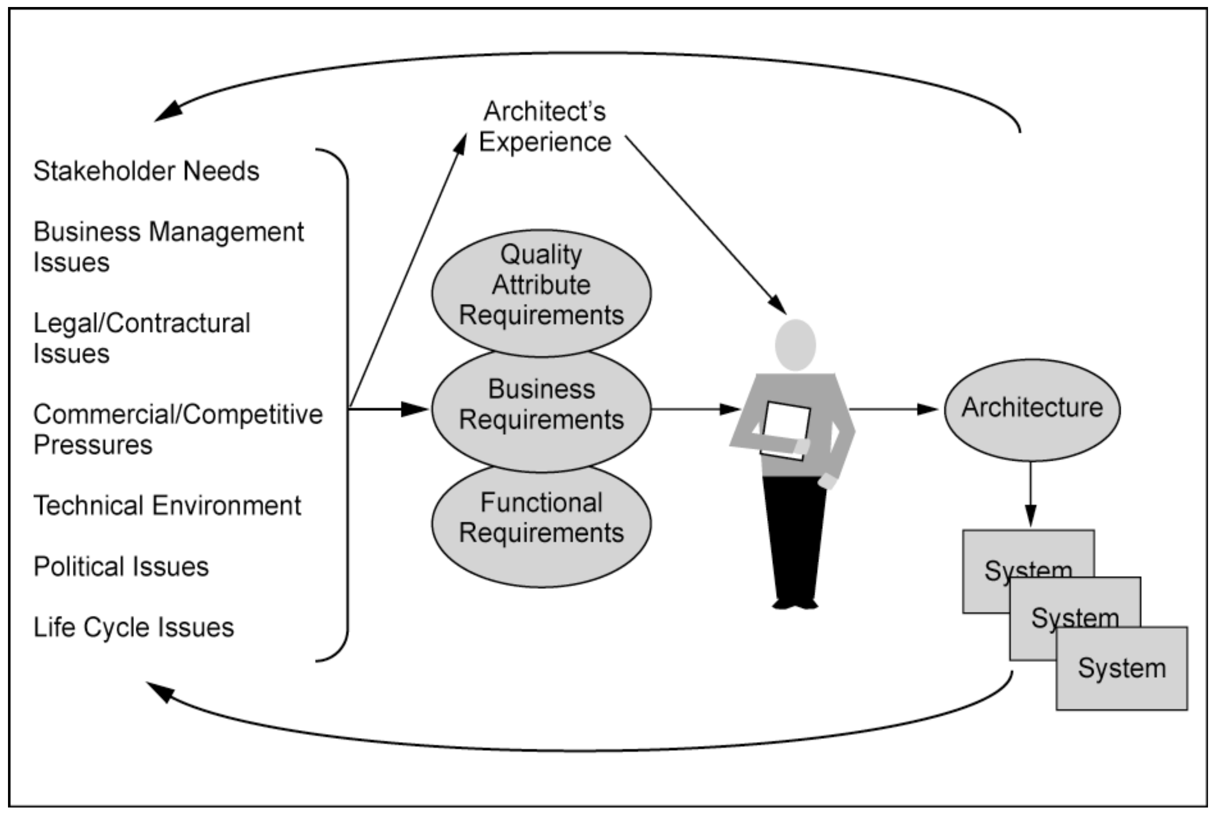
Software Engineering Institute

STILL MORE DEFINITIONS

> *A formal description of a system, or a detailed plan of the system at component level to guide its implementation. The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time.*

Open Group Architecture Framework

WHO CARES ABOUT ARCHITECTURE?



Forces that influence architecture (Image credit: SEI)

MANY STAKEHOLDERS, MANY PERSPECTIVES

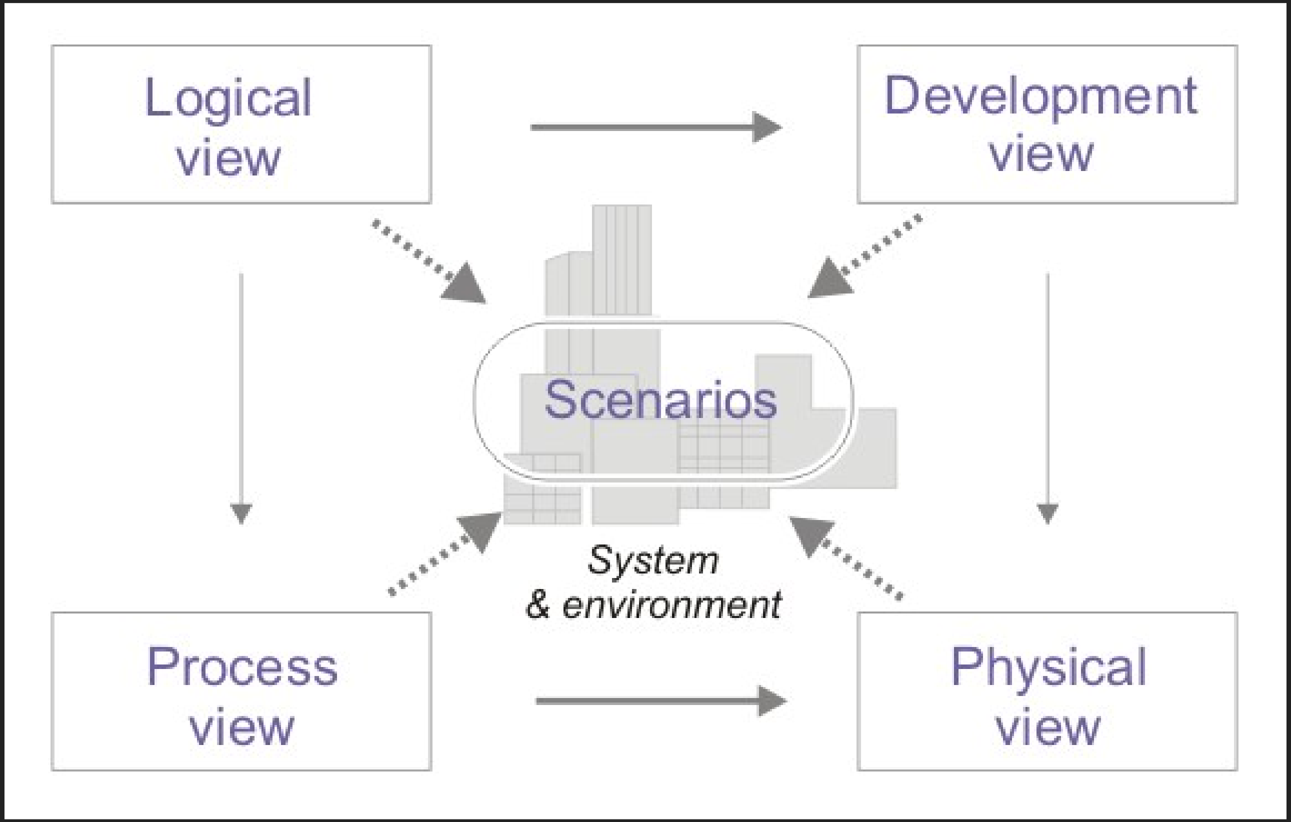
Architecture views are representations of the overall architecture that are meaningful to one or more stakeholders in the system.

The architect chooses and develops a set of views that will enable the architecture to be communicated to, and understood by, all the stakeholders, and enable them to verify that the system will address their concerns.

MANY DIFFERENT VIEWS

Stakeholders	Views	Perspectives
Users, Biz Management	Business View	Flow of business information
Database Designers	Data View	Flow of organizational data
Software & System Engineers	Applications View	Development of code & integration
Acquirers, Administrators	Technology View	COTS, hardware

STILL MORE VIEWS



Philippe Kruchten's 4+1 view of architecture

HOW DOES IT RELATE TO DESIGN?

> *All architecture is design but not all design is architecture. Architecture represents the significant design decisions that shape a system, where significant is measured by cost of change.*

Grady Booch in *Pattern-Oriented Software Architecture, On Patterns and Pattern Languages*

SIGNIFICANT COST OF CHANGE

- Security
- Scalability
- Availability
- Privacy

ARCHITECTURAL CONSIDERATIONS -- SECURITY

Encryption:

- Do transactions need to be encrypted?
- What's the legal minimum required? (Banks, Health, Govt)

User Identification:

Cookies

Certificates

Bio-metrics

ARCHITECTURAL CONSIDERATIONS - AVAILABILITY

- How much downtime is acceptable?
- How much data loss is tolerable?
- In a crisis, what is the target time for resuming service?

ARCHITECTURAL CONSIDERATIONS - SCALABILITY

- How much additional traffic can the system handle?
- How easy is it to add capacity?

Deployment:

- How many servers will be required?
- Are these servers geographically separated?
- Is the infrastructure owned by the company? Leased?

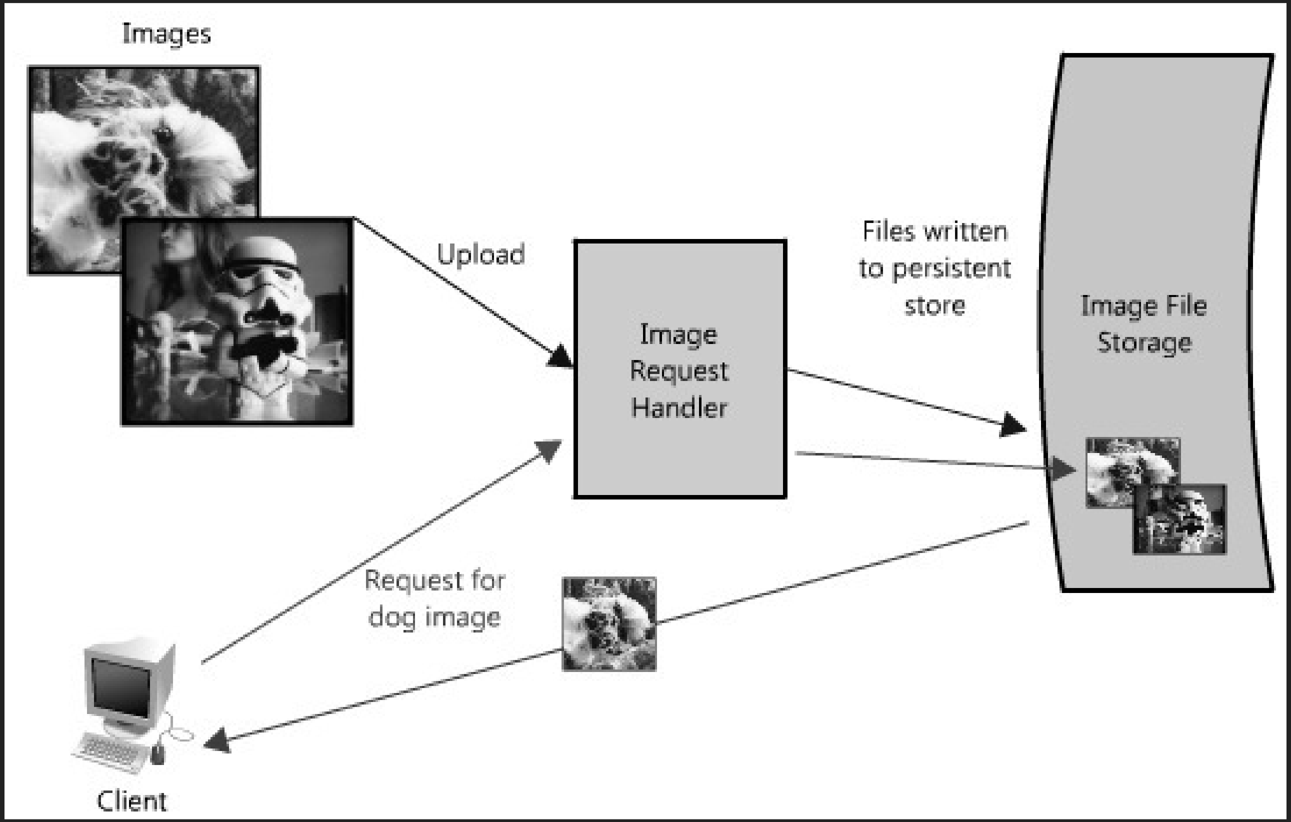
ALL ABOUT THE MONEY, HONEY!

The answers to all of the questions in the previous slides cost money. Significant money.

CASE STUDY - AN INTERNET-SCALE APPLICATION

Image/Video sharing application (Flickr, Facebook, Imgur,...)

ONE POSSIBLE CANDIDATE



WHAT CHARACTERISTICS ARE IMPORTANT?

- Facebook has about 400 *petabytes* of data (and growing every day)
- More than 300 billion images (in 2015)
- About 350 million photos added every day

STAKEHOLDERS DEMAND: FUNCTIONALITY

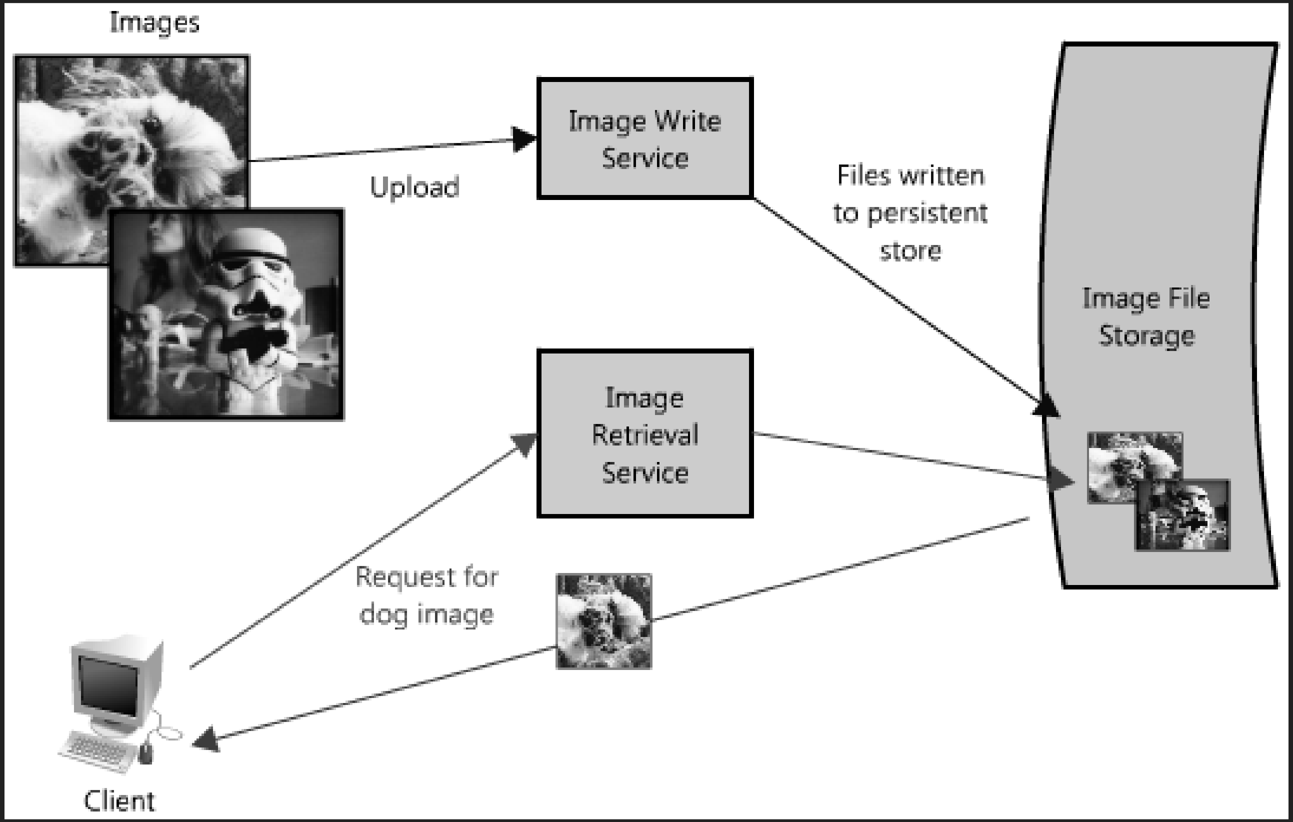
- Low latency for image downloads/requests (performance)
- Never (never, never) lose data (redundancy)
- No limit on number of images (scalability)

DECISIONS, DECISIONS, DECISIONS

Constraints:

- Decouple functionality (think service-oriented architecture)
- Typical IP networks are built for 3:1 (download:upload) speed
- Apache, lighthttpd, nginx all have a maximum connection limit

MODIFY THE EARLIER ARCHITECTURE

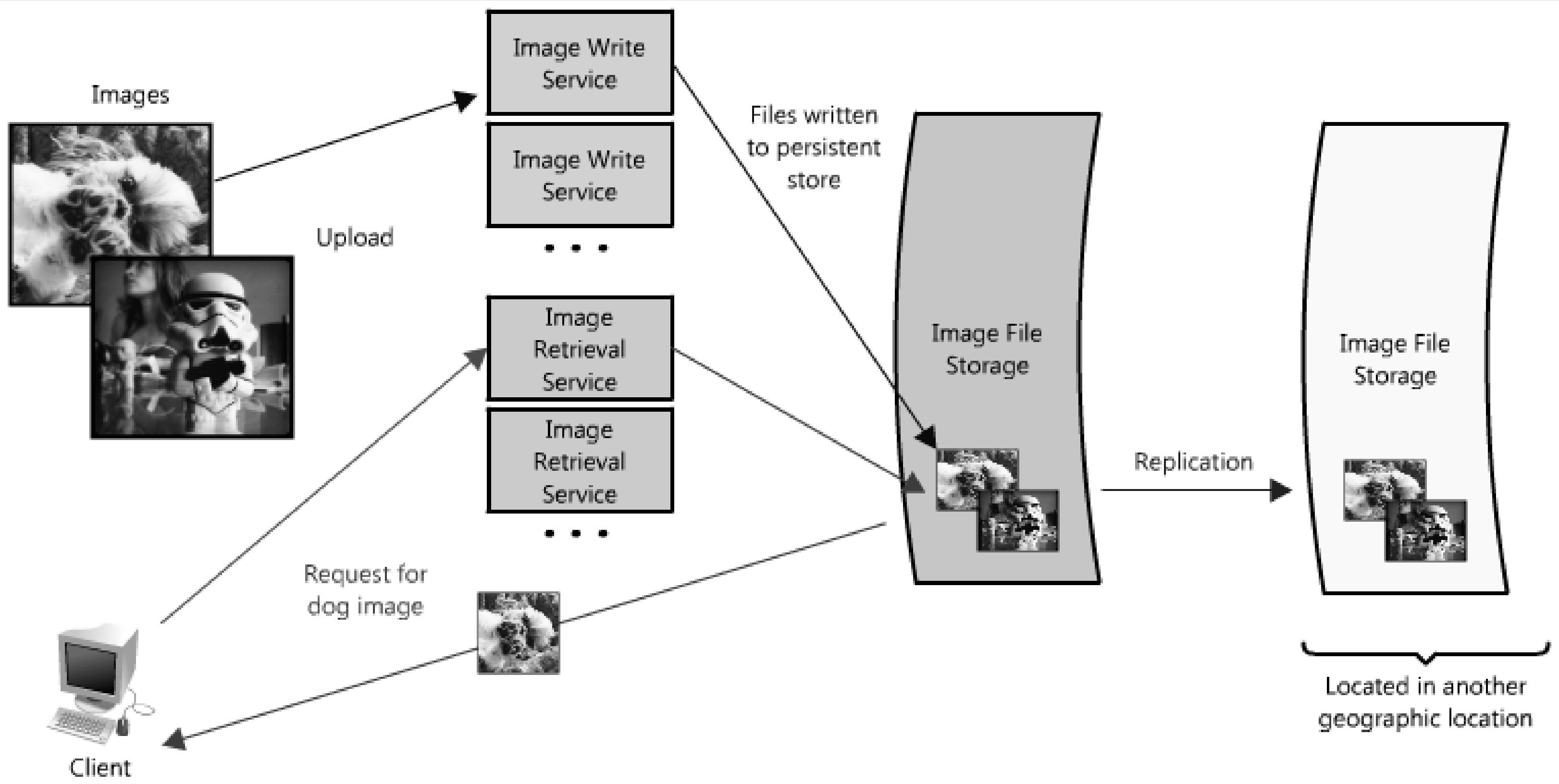


GO BACK TO THE STAKEHOLDERS' CONCERNS

Redundancy:

- Remove single point of failure
- Multiple copies with graceful switchover
- *Shared-nothing* style of architecture

MODIFY AGAIN

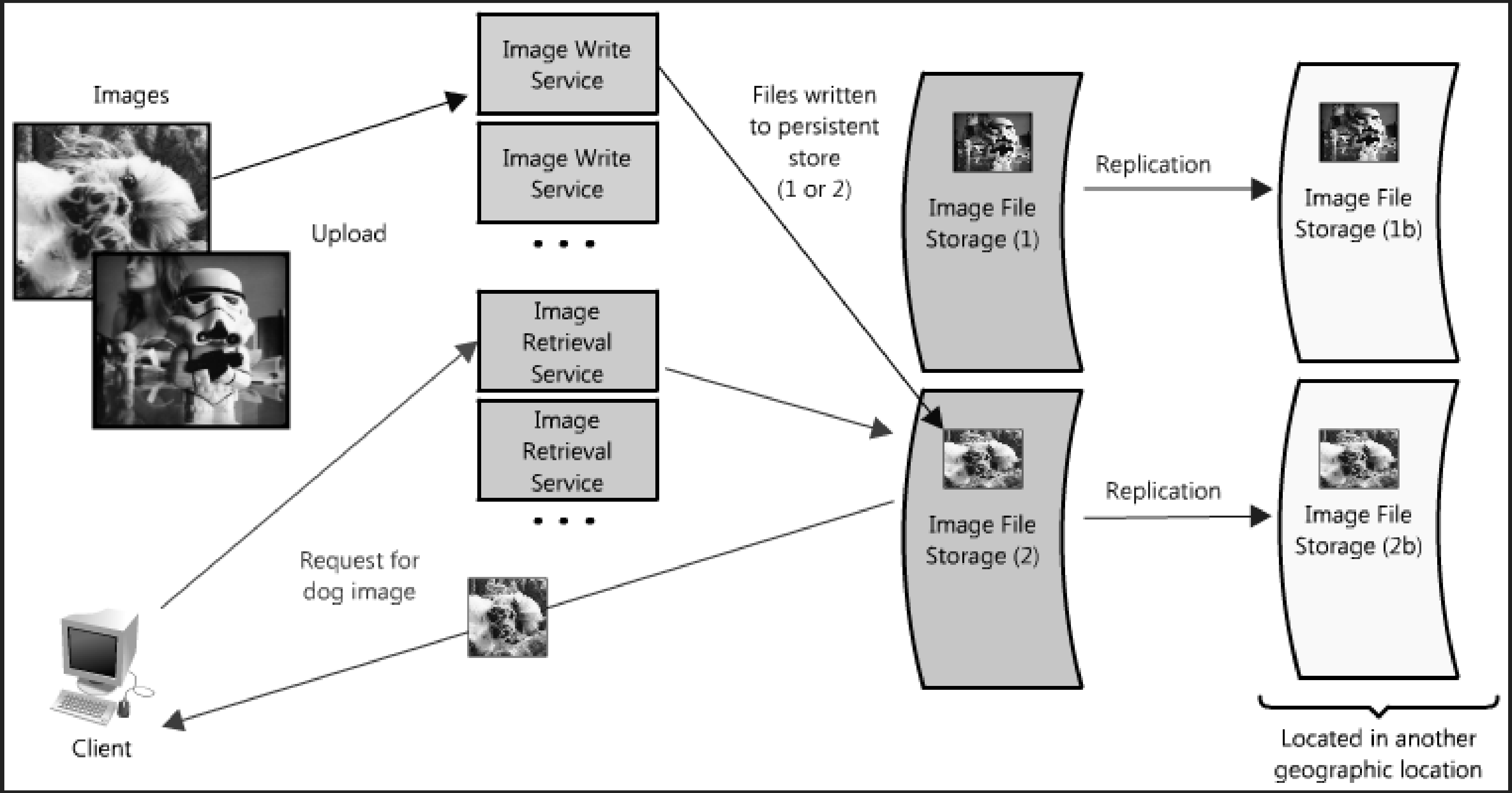


BACK TO STAKEHOLDERS' CONCERNS

No limit on number of images:

- Too many images to fit on a disk
- Scale vertically (add more disks, more CPU, more RAM, ...)
- Scale horizontally (partition data on to multiple servers)

MODIFY YET AGAIN



DOES ARCHITECTURE IMPACT DESIGN?

Database design, API design:

- Given a request, how do you know which disk the image is on?
- How do you maintain indexes? What about tags attached to images?

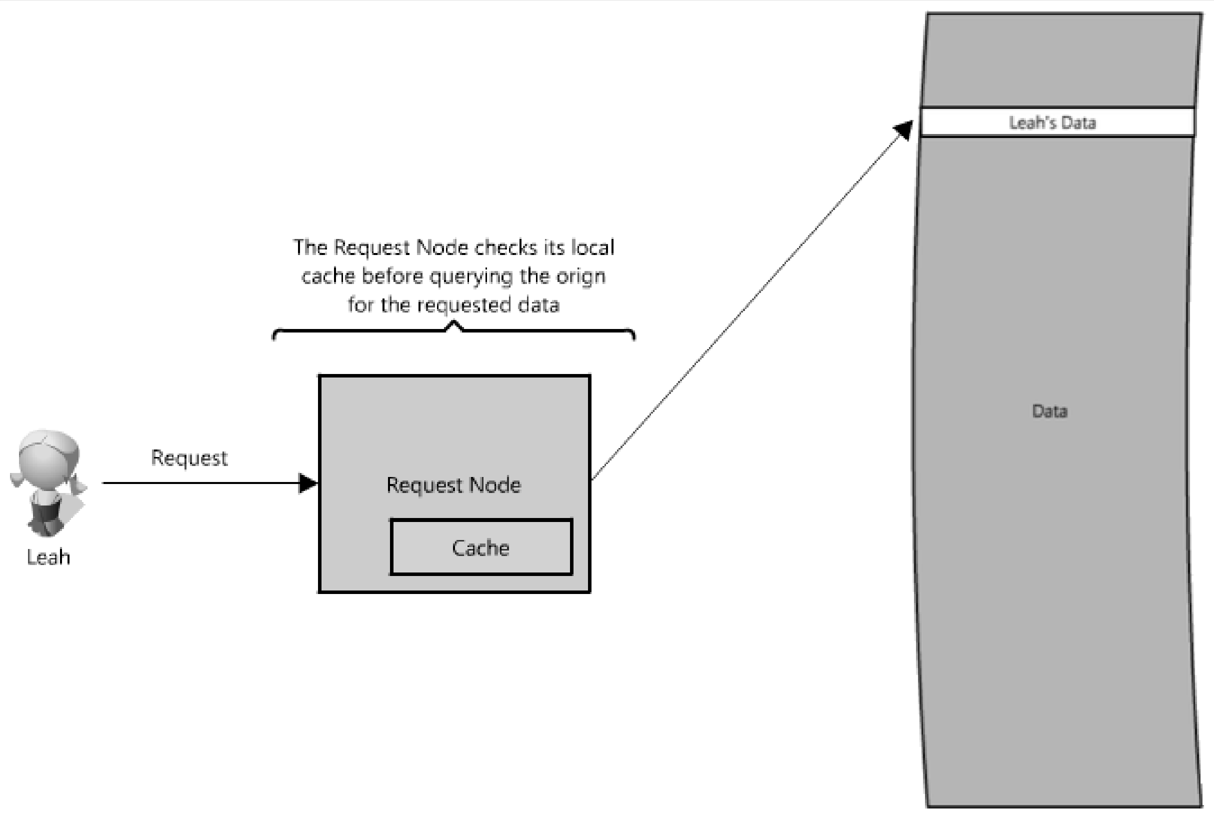
LOTS MORE ISSUES

What happened to *latency*? [performance]

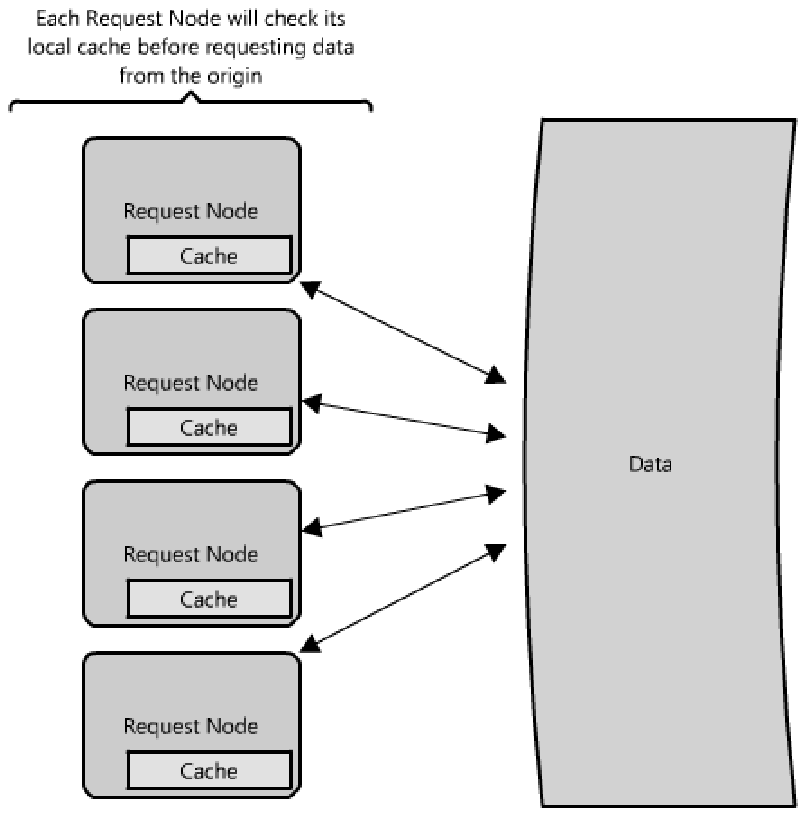
What happened to *consistency*? [tracking]

What happened to *cost* ?

WHEN YOU TRY TO SOLVE ONE ISSUE ...


The Request Node checks its local cache before querying the orign for the requested data
Request
Request Node
Cache
Leah
Leah's Data
Data

YOU NEGATIVELY IMPACT THE OTHER


Each Request Node will check its local cache before requesting data from the origin
Request Node
Cache
Request Node
Cache
Request Node
Cache
Request Node
Cache
Data

ARCHITECTURE IS ABOUT TRADEOFFS

Balancing concerns:

- Each stakeholder prioritizes his/her concerns
- The architect needs to balance all of them
- Never be a purist; mix-n-match styles to achieve goals

NEXT CLASS: ARCHITECTURAL STYLES

THAT'S ALL, FOLKS!

Questions? Comments?