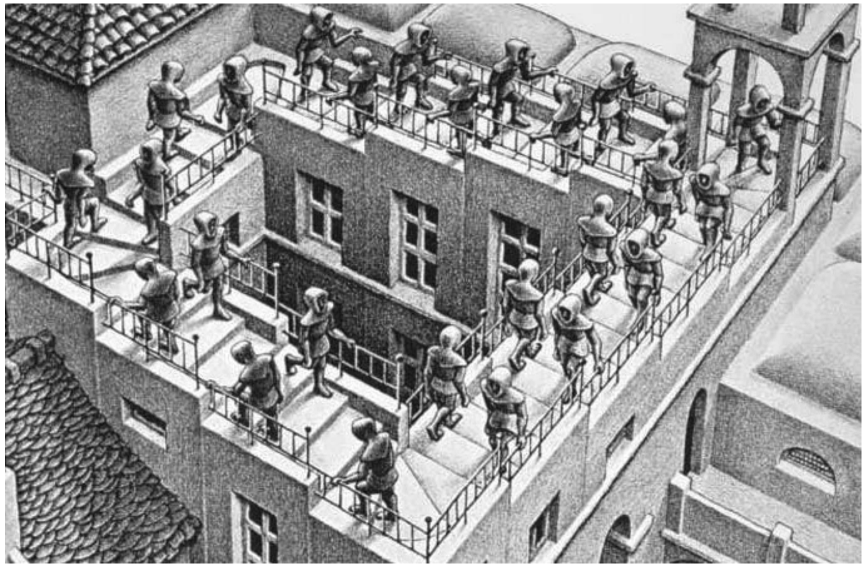
Case Studies



Case Study #1: From Mainframe To…

- 64% of organizations still use mainframe-based applications that are between 10 and 20 years' old, 28% are between 20 and 30 years' old
- mantra "If it ain't broke, don't fix it"
- modernization activity is a key part of improving 70% of organizations' carbon footprint
- the Cloud is cheaper to operate than mainframes for 60% of organizations

- 36% of organizations consider the legacy modernization programs they have completed, to be failures

- What transitioning strategies are possible?

https://modernsystems.oneadvanced.com/globalassets/modern-systems-assets/resources/reports/advanced_mainframe_report_2021.pdf

Case Study #1: From Mainframe To…

- Documents:
  - *A Graceful Modernization Journey*

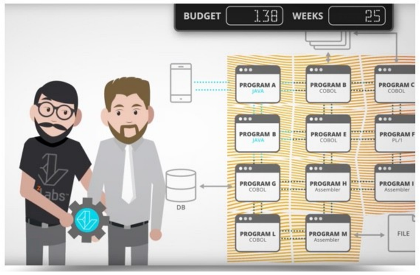
    LzLabs SDM - Dec. 2018

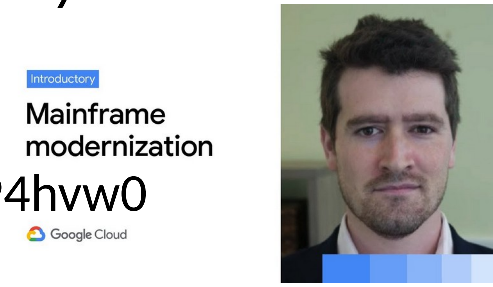
    Video: https://www.youtube.com/watch?v=4nlv5Fw0wd4
  - *Mainframe modernization: Accelerating legacy transformation (7:00 -> 13:00)*

    Google G4 - Sept. 2020

    Video: https://www.youtube.com/watch?v=-er5J94hvw0
- Proposal:

  Present and contrast the 2 reported approaches for gradually transitioning from a mainframe architecture





Case Study #2: From Monolith to Microservices

- Working with a monolith makes it very challenging to onboard new developers into the team, as they will spend months learning the system's codebase before being able to start working on it or being productive.
- Even the most skilled development teams hesitate to make changes or add new code that might disrupt the system's operation in unexpected ways.

- How to enable new collaborators to develop microservices rather than diving into the monolith?

Case Study #2: From Monolith to Microservices

- Document:

  From Monolith to Microservices

  Sha Ma (Vice President of Software Engineering at GitHub)

  Qcon Plus Conference, April 2021

  Video presentation (from 5:14) and transcript (from “be pragmatic”): https://www.infoq.com/presentations/github-rails-monolith-microservices/?topicPageSponsorship=dca0fcc9-a580-4af1-870b-9be845a6780f&itm_source=presentations_about_architecture-design&itm_medium=link&itm_campaign=architecture-design

- Proposal:

  Summarize the process adopted to transition from the monolith to microservices

Qcon 2020

Monolith to Microservices

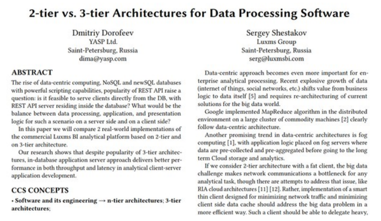
Presented by @shama0 | VP, Software Engineering @ GitHub

Case Study #3:
2-tier vs. 3-tier Architectures

- When designing and implementing a data analysis application, there are multiple candidate architectures, such as a 2-tier architecture (DB with stored procedures) or a 3-tier architecture (separate DB and application server).

- To what extent is one strategy better than the other?

Case Study #3: 2-tier vs. 3-tier Architectu



2-tier vs. 3-tier Architectures for Data Processing Software

Dmitriy Dorofeev
YASP Ltd.
Saint-Petersburg, Russia
dima@yasp.com

Sergey Shestakov
Luxms Group
Saint-Petersburg, Russia
serg@luxmsbi.com

ABSTRACT

The rise of data-centric computing, NoSQL and newSQL databases with powerful scripting capabilities, popularity of REST API raise a question: is it feasible to serve clients directly from the DB, with REST API server residing inside the database? What would be the balance between data processing, application, and presentation logic for such a scenario on a server side and on a client side?

In this paper we will compare 2 real-world implementations of the commercial Luxms BI analytical platform based on 2-tier and on 3-tier architecture.

Our research shows that despite popularity of 3-tier architectures, in-database application server approach delivers better performance in both throughput and latency in analytical client-server application development.

CCS CONCEPTS

• **Software and its engineering → n-tier architectures; 3-tier architectures;**

Data-centric approach becomes even more important for enterprise analytical processing. Recent explosive growth of data (internet of things, social networks, etc.) shifts value from business logic to data itself [5] and requires re-architecturing of current solutions for the big data world.

Google implemented MapReduce algorithm in the distributed environment on a large cluster of commodity machines [2] clearly follow data-centric architecture.

Another promising trend in data-centric architectures is fog computing [1], with application logic placed on fog servers where data are pre-collected and pre-aggregated before going to the long term Cloud storage and analytics.

If we consider 2-tier architecture with a fat client, the big data challenge makes network communications a bottleneck for any analytical task, though there are attempts to address that issue, like RIA cloud architectures [11] [12]. Rather, implementation of a smart thin client designed for minimizing network traffic and minimizing client side data cache should address the big data problem in a more efficient way. Such a client should be able to delegate heavy,

- Document:

  2-tier vs. 3-tier Architectures for Data Processing Software

  Dmitriy Dorofeev (YASP Ltd.) and Sergey Shestakov (Luxms Group)

  International Conference on Applications in Information Technology (ICAIT), November 2018

  Article: https://dl.acm.org/doi/10.1145/3274856.3274869

- Proposal:

  Provide a summary table of observations and conclusions (tip: skim through sections 1 to 4 and then divide the work: study sections 5 and 6 of the article in parallel).

Case Study #4: To Microservices and Back Again

- As software systems grow, major refactoring is sometimes needed and transitioning from the monolith to the microservice architecture is appealing.

- Can one dive in headfirst?

Case Study #4: To Microservices and Back Again

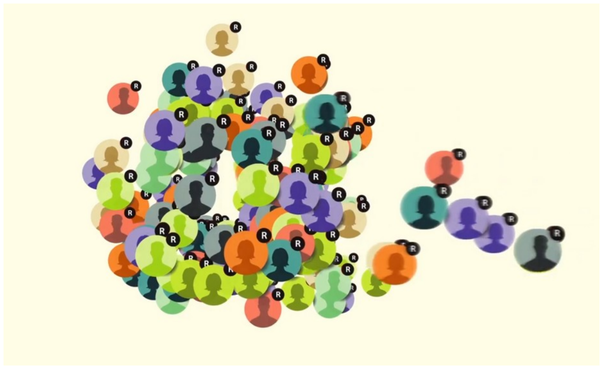
- Document:
  - To Microservices and Back Again
  - Alexandra Noonan (Segment)
  - Video (recorded at QCon 2020): https://www.youtube.com/watch?v=hIFeaeZ9_AI
  - Article: https://segment.com/blog/goodbye-microservices/

- Proposal:
  - What was the initial motivation for introducing microservices?
  - Where did it go wrong?

Case Study #5: Broadcasting live to millions

- a video streaming broadcast server
- Millions of clients worldwide attempt to watch the same video simultaneously

- How to prevent server saturation?



Case Study #5: Broadcasting live to millions

- Document:
  - https://www.facebook.com/Engineering/videos/10153675295382200
  - https://engineering.fb.com/2015/12/03/ios/under-the-hood-broadcasting-live-video-to-millions/
- Proposal:
  - Present the set of solutions set up by Facebook

Case Study #6: CQRS

- CQRS is an application architecture that separates the commands (write operations) from the queries (read operations)