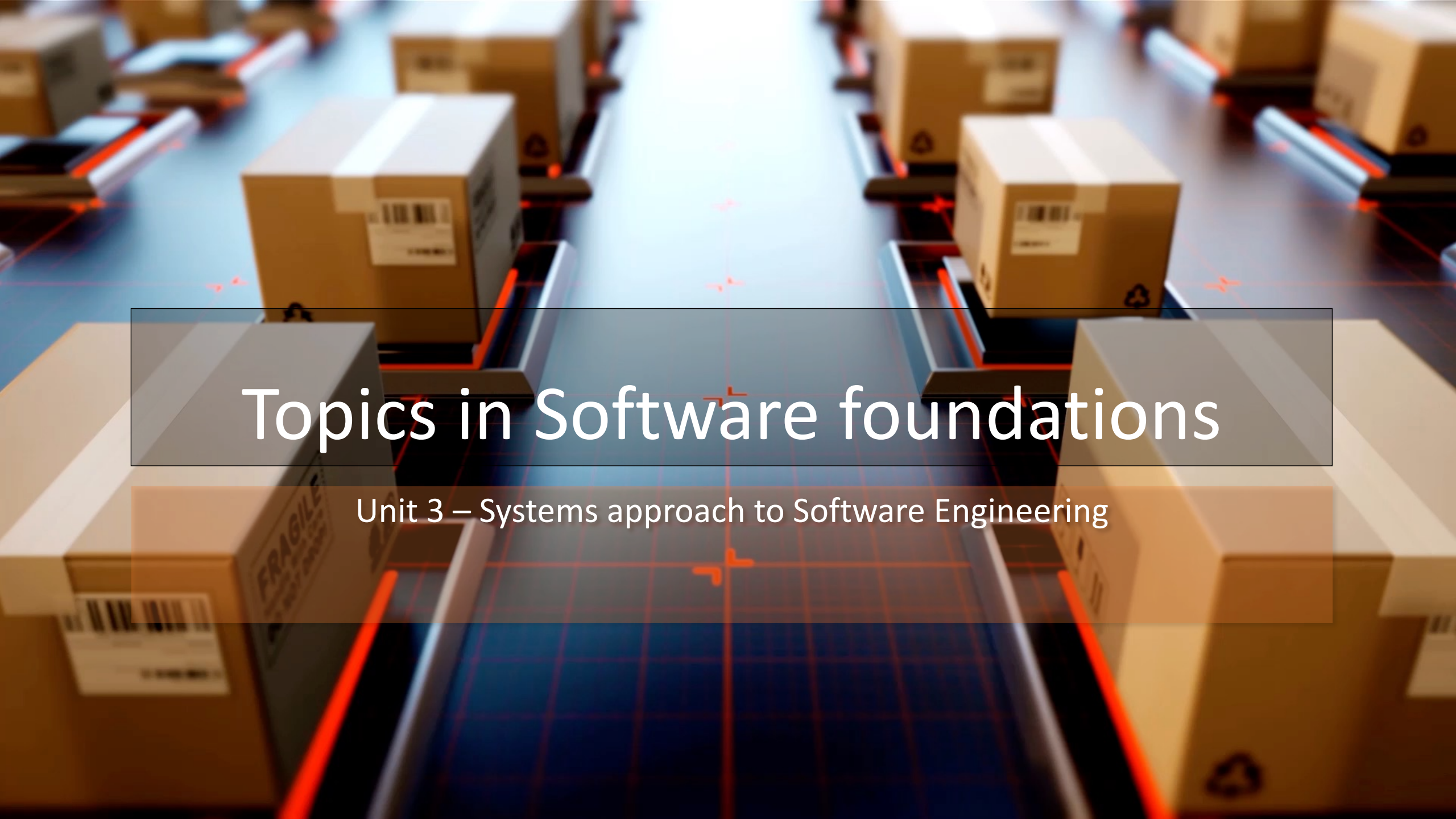


Topics in Software foundations

Unit 3 – Systems approach to Software Engineering

Unit 3 Learning Objectives

- Understand the concept of software modeling (distinct from architecture and design)
- Understand System of Systems in terms of component systems and their interactions
- Apply transition systems to model software systems and System of Systems
- Create a model for a software product given the running system
- Create a model for a software product given the specification

Topics we cover

- Introduction to systems modeling
- Modeling System of Systems (SoS)
- Modeling communications
- Modeling UI systems
- Modeling data access
- Formalizing TS as a modeling language

Modeling Software Systems

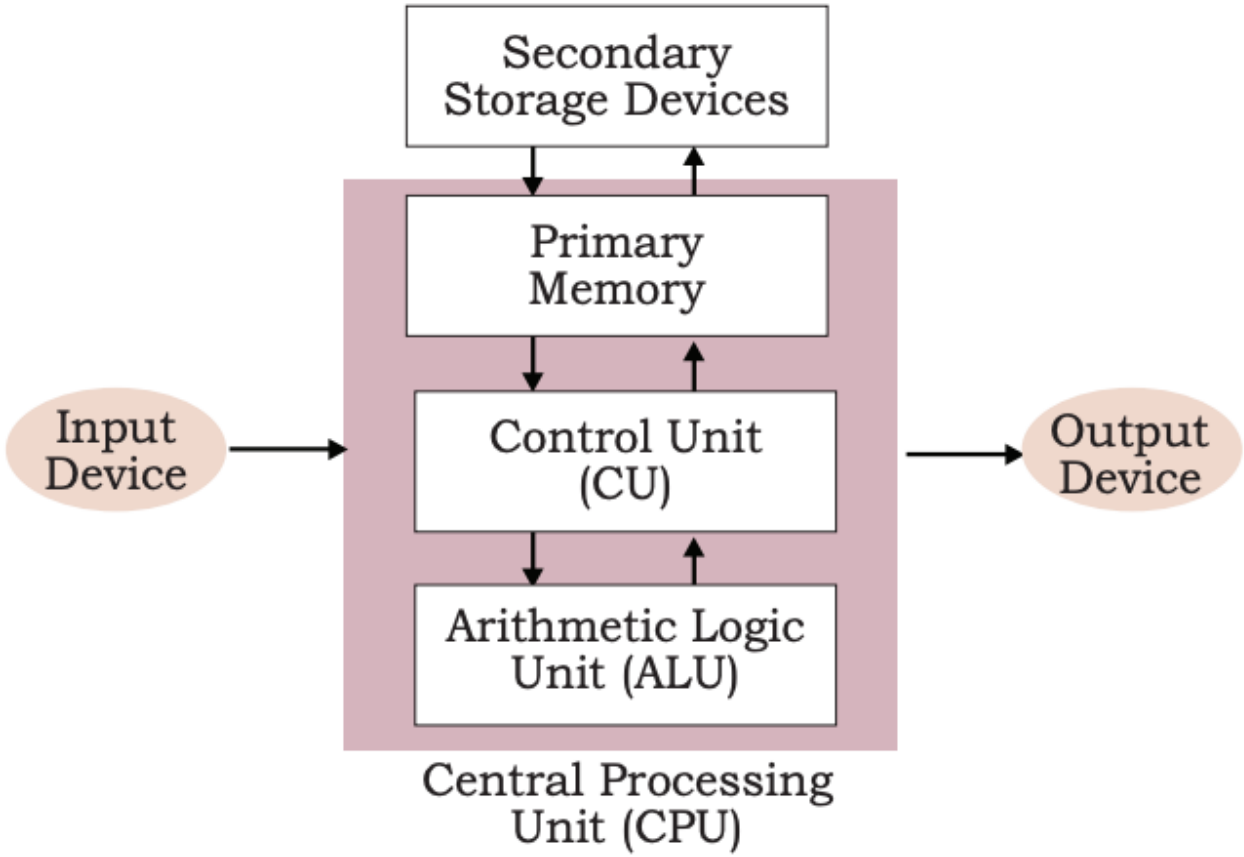
Unit 3 – TiSF S'23

Session 2 (2023-03-27)

Model

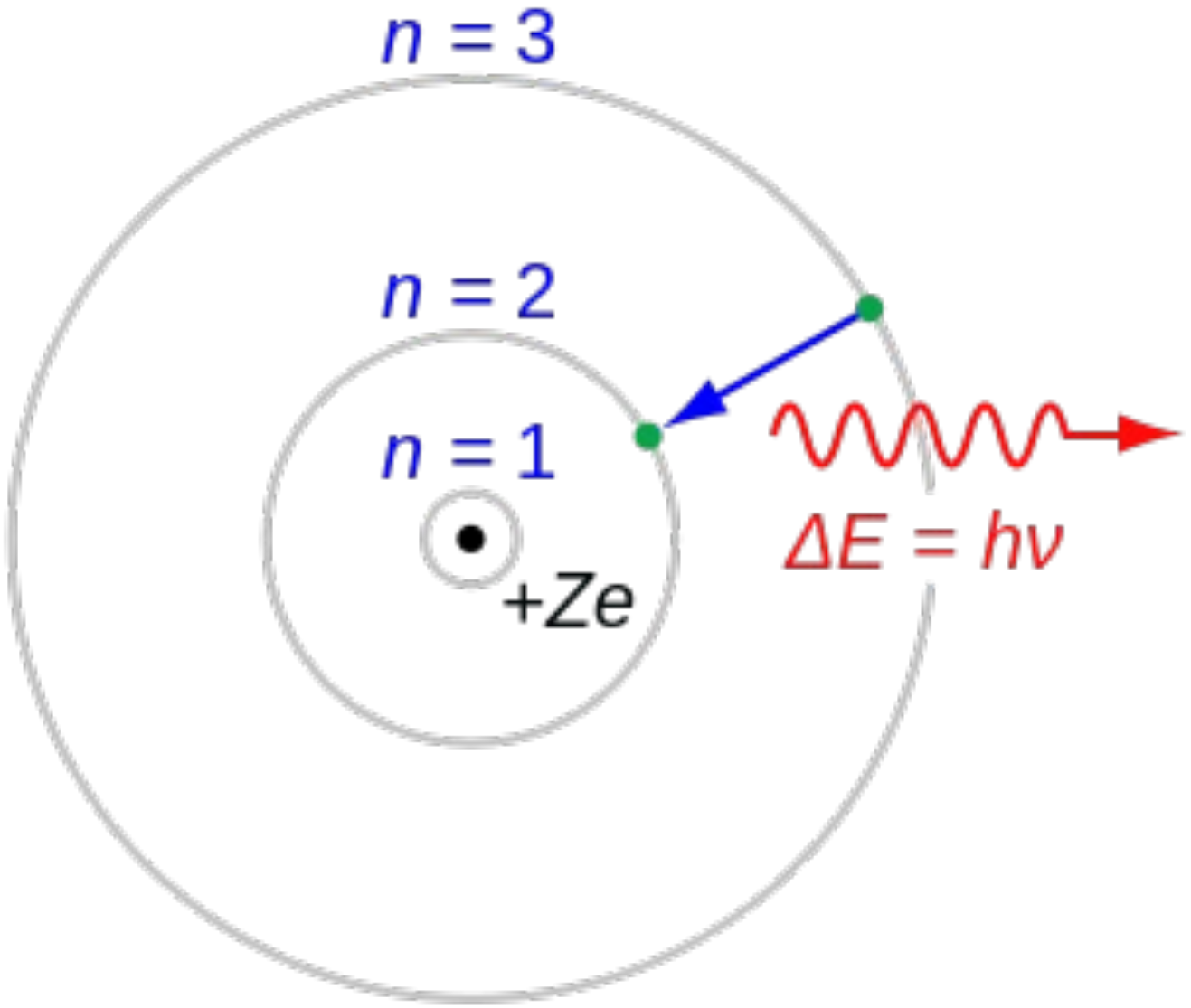
A theoretical representation of a system (or of a part of it) at the desired level of abstraction so that it can be be understood, analyzed or simulated

Model of the computer

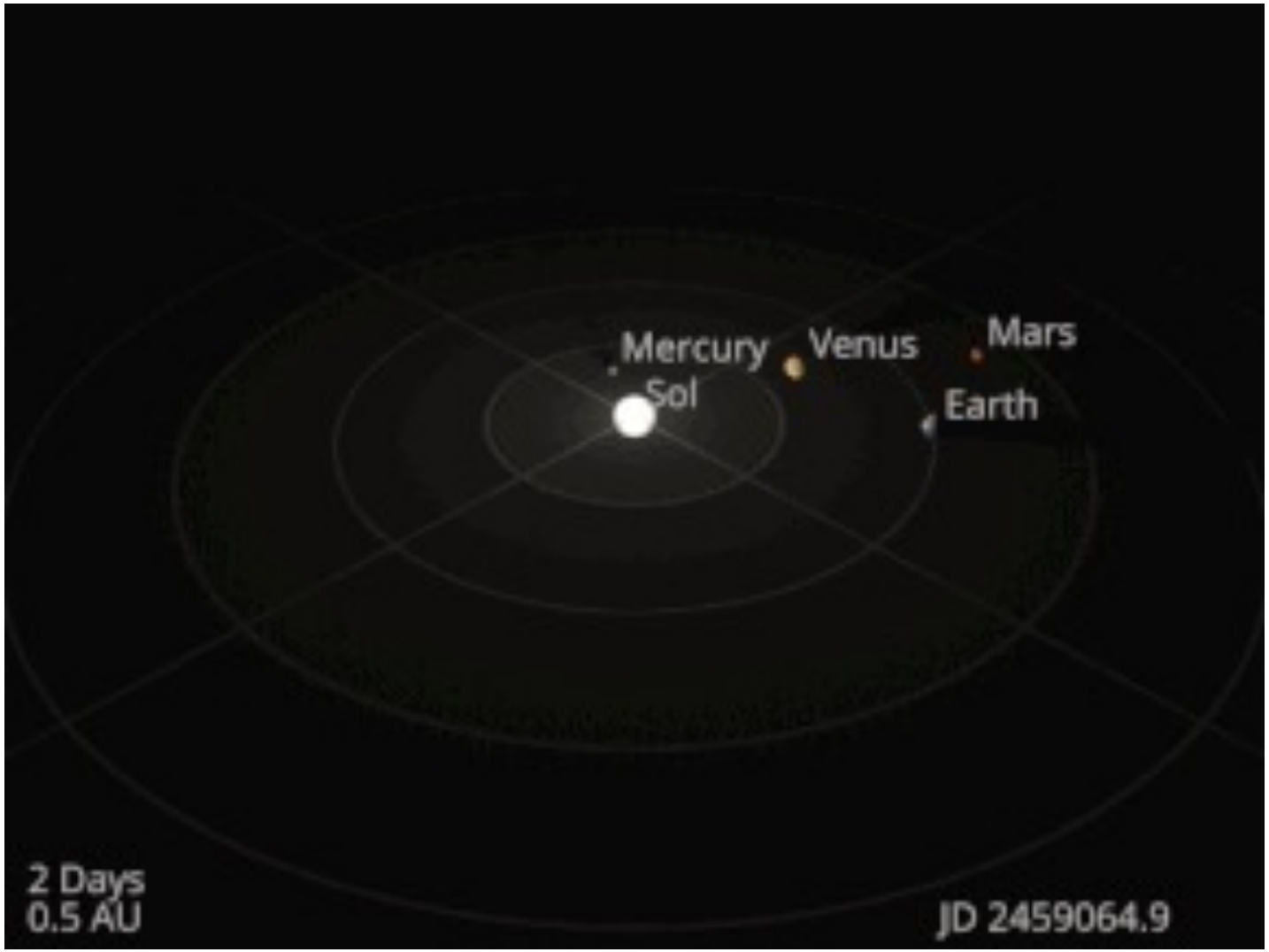


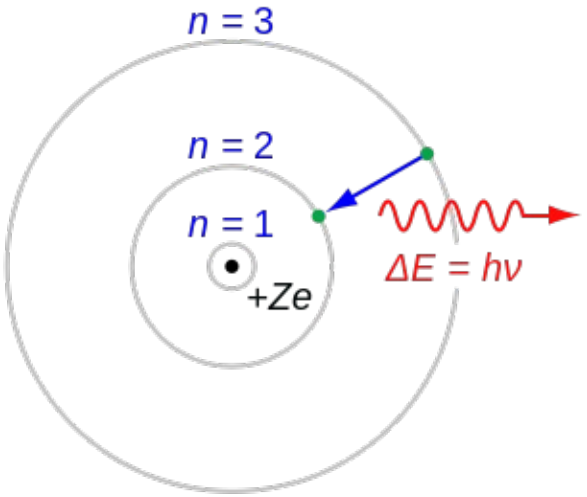
Model of the Earth

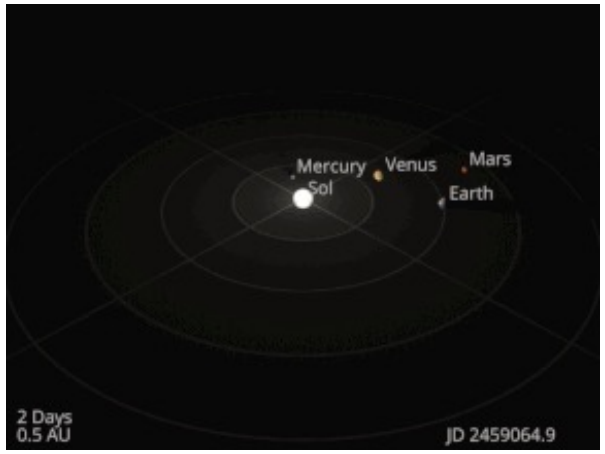
Model of the atom

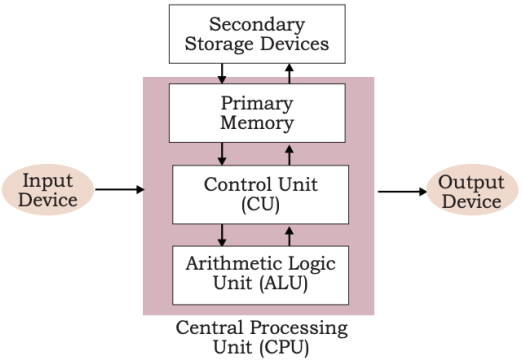


Model of the Solar system









We use models all the time to understand behaviors of systems

Image credits: Wikipedia, NCERT

Different models for the same system

Calculator	Car
• For a 3th grader who just saw one • For a 10th grader who started learning programming • For a CSE undergrad student who learned UI development • For an EE undergrad who learned circuits and systems	• To show the locomotion aspects • To show aesthetics • To show mechanics

Let’s model this classroom

A model should cover one or more of

Requirements	Structure	Interconnections	Dynamics/Behavior
• Functional and behavioral expectations from the system	• Components and sub-systems that form the system	• Connections between various components and systems	• The rules governing the response of the system in the presence of a stimulus

Is software systems modeling different?

A Factorial system

Requirements	Structure	Interconnection	Dynamics/Behavior
• When n is provided, the result should be n! for all n > 0. • System should abort if it takes more than 10 ms for a particular n	• A multiplication unit (S1) • A counter (S2) • Input/output handler (S3)	• S3→S2→S1→S3	• Get the number (S3) and reset the accumulator to 1 • Count from 1 to n (S2) and multiply each number with the accumulated product (S1) • Show result (S3)

A software model should enable (by hand or by tool)

Comprehension

- Ease of understanding and visualizing

Simulation

- Tracing the behavior over (discrete) time as predicted by the model

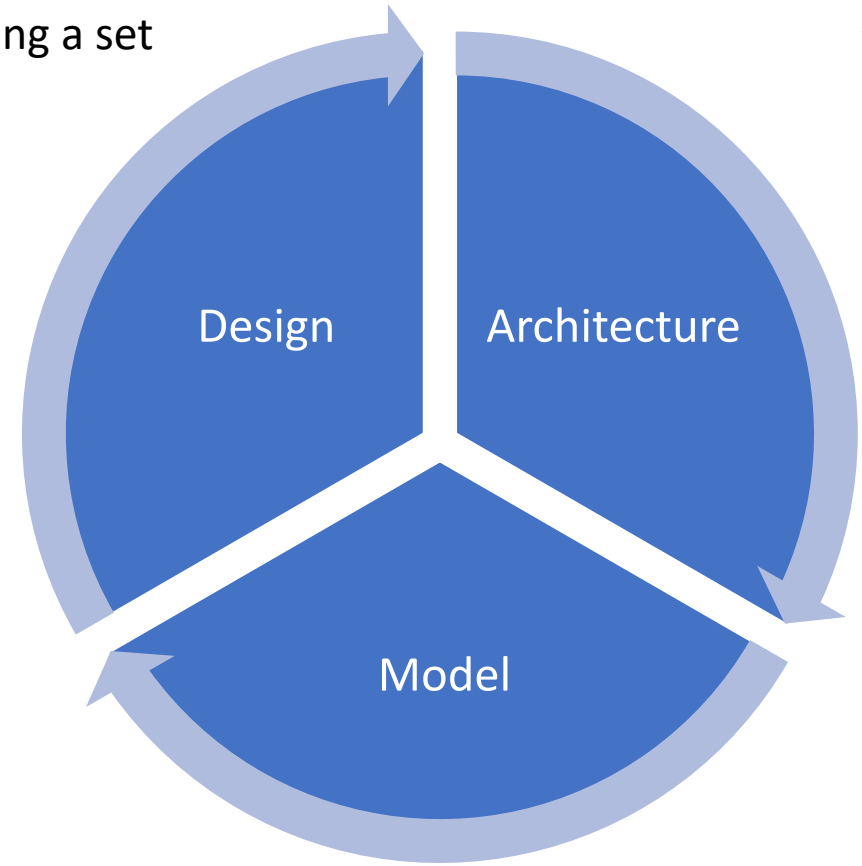
Verification and Validation

- Ability to show that the model conforms to requirements (verification) and to the user expectations (validation)

Do we need to distinguish amongst these terms?

Design
Software design is the process by which an agent creates a specification of a software artifact intended to accomplish goals, using a set of primitive components and subject to constraints.

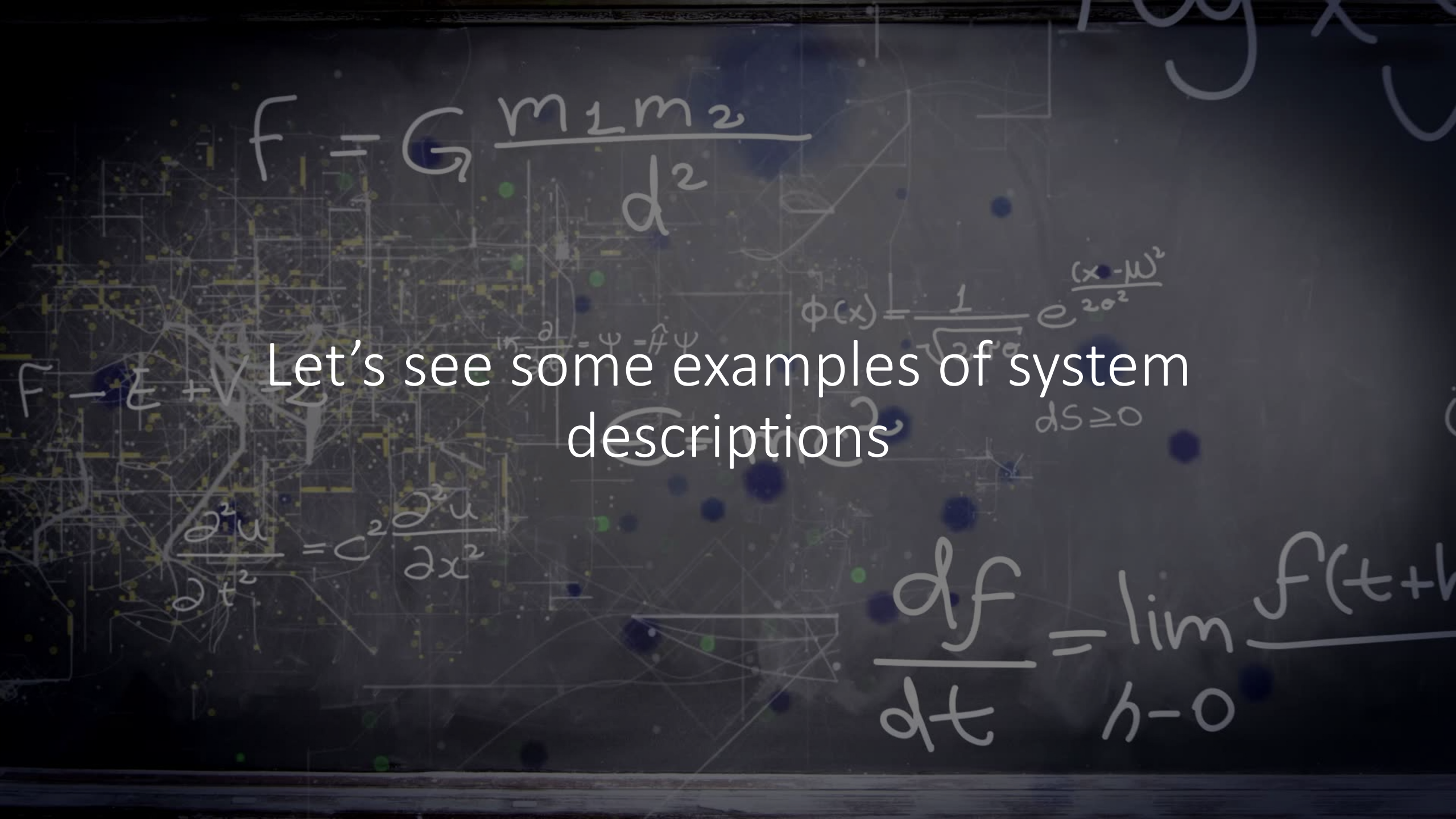
- Wikipedia



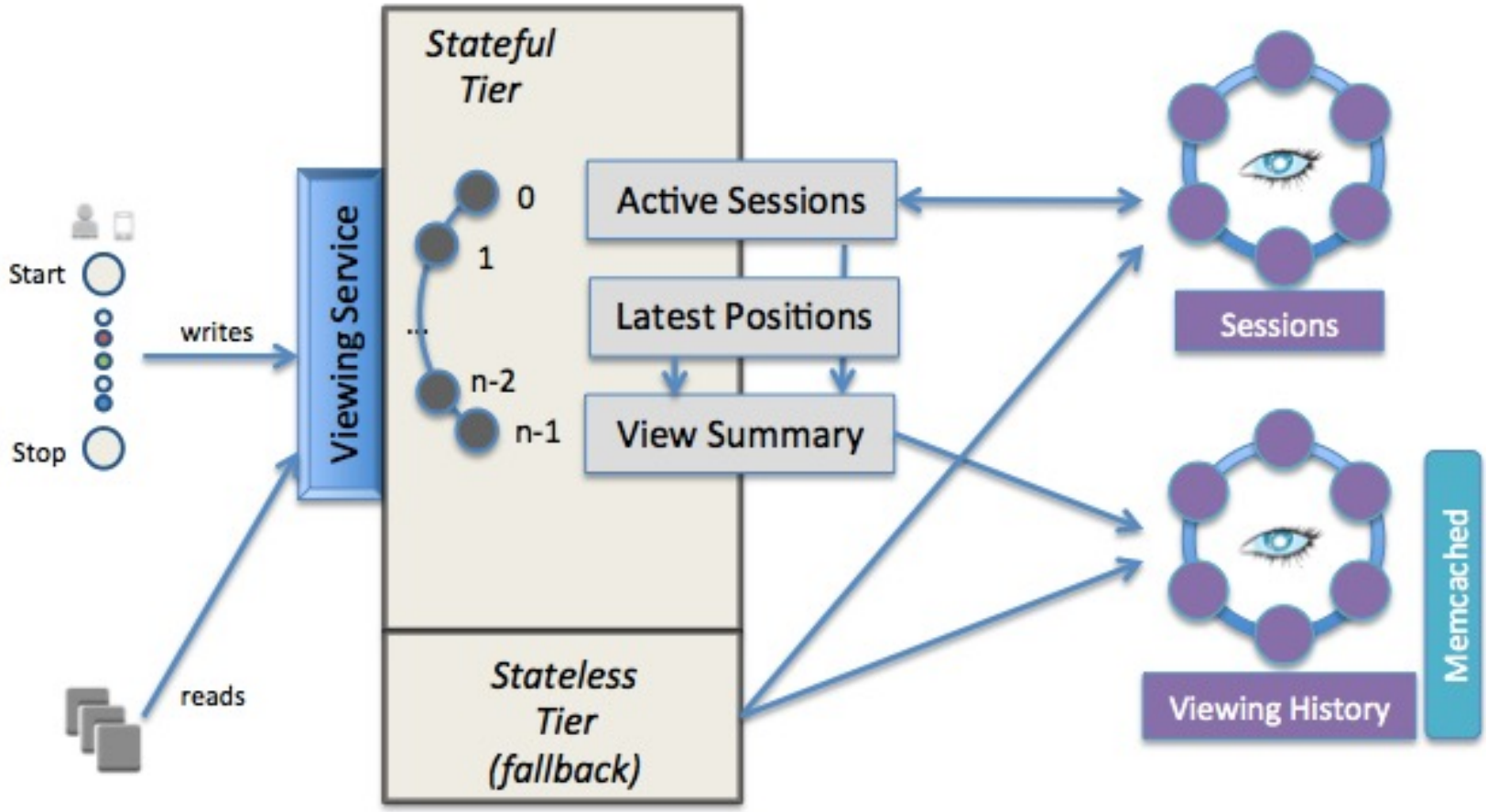
Architecture
The software architecture of a system represents the design decisions related to overall system structure and behavior. Architecture helps stakeholders understand and analyze how the system will achieve essential qualities such as modifiability, availability, and security.

- CMU

Different ways of describing a system serving different purposes

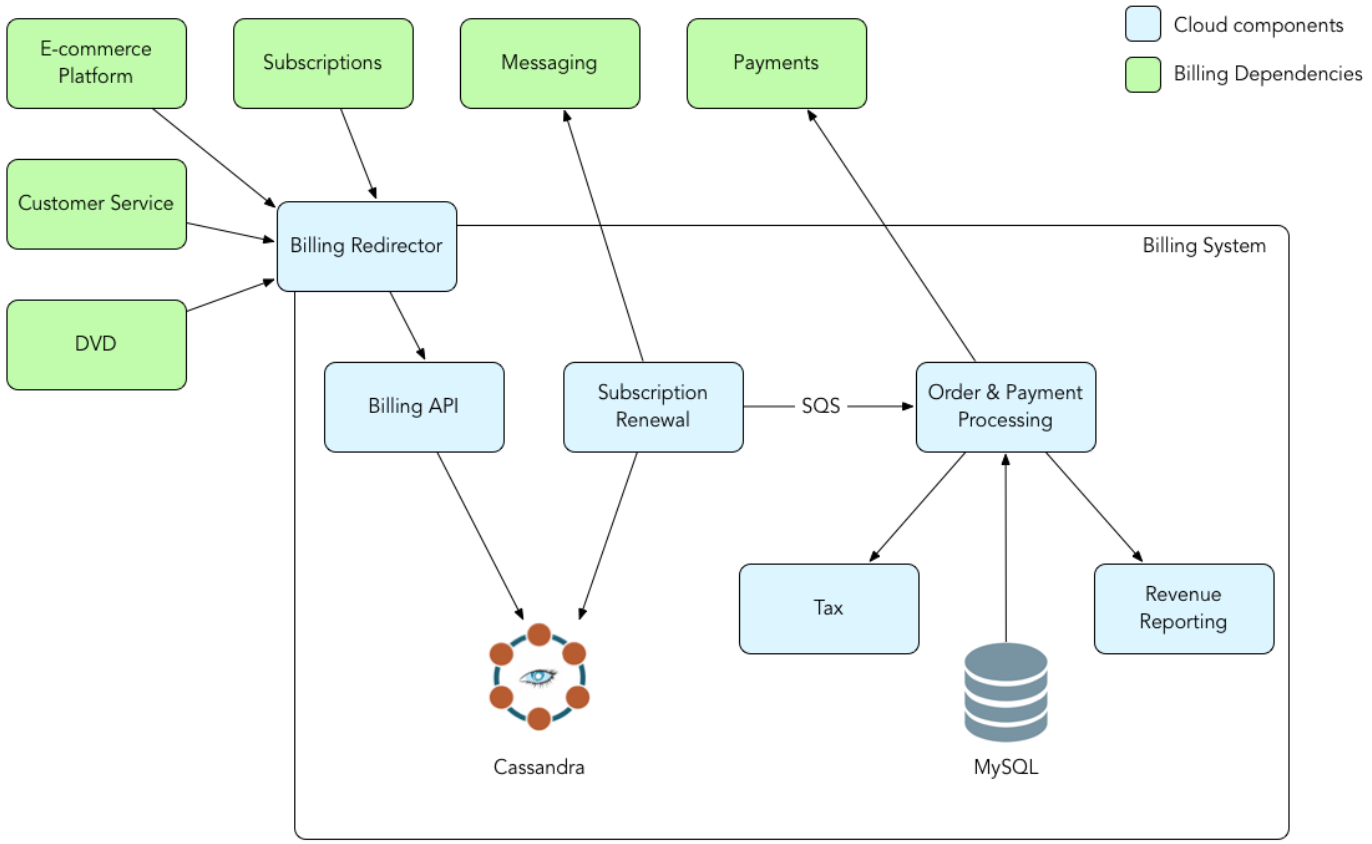

Let’s see some examples of system descriptions

Netflix Viewing data



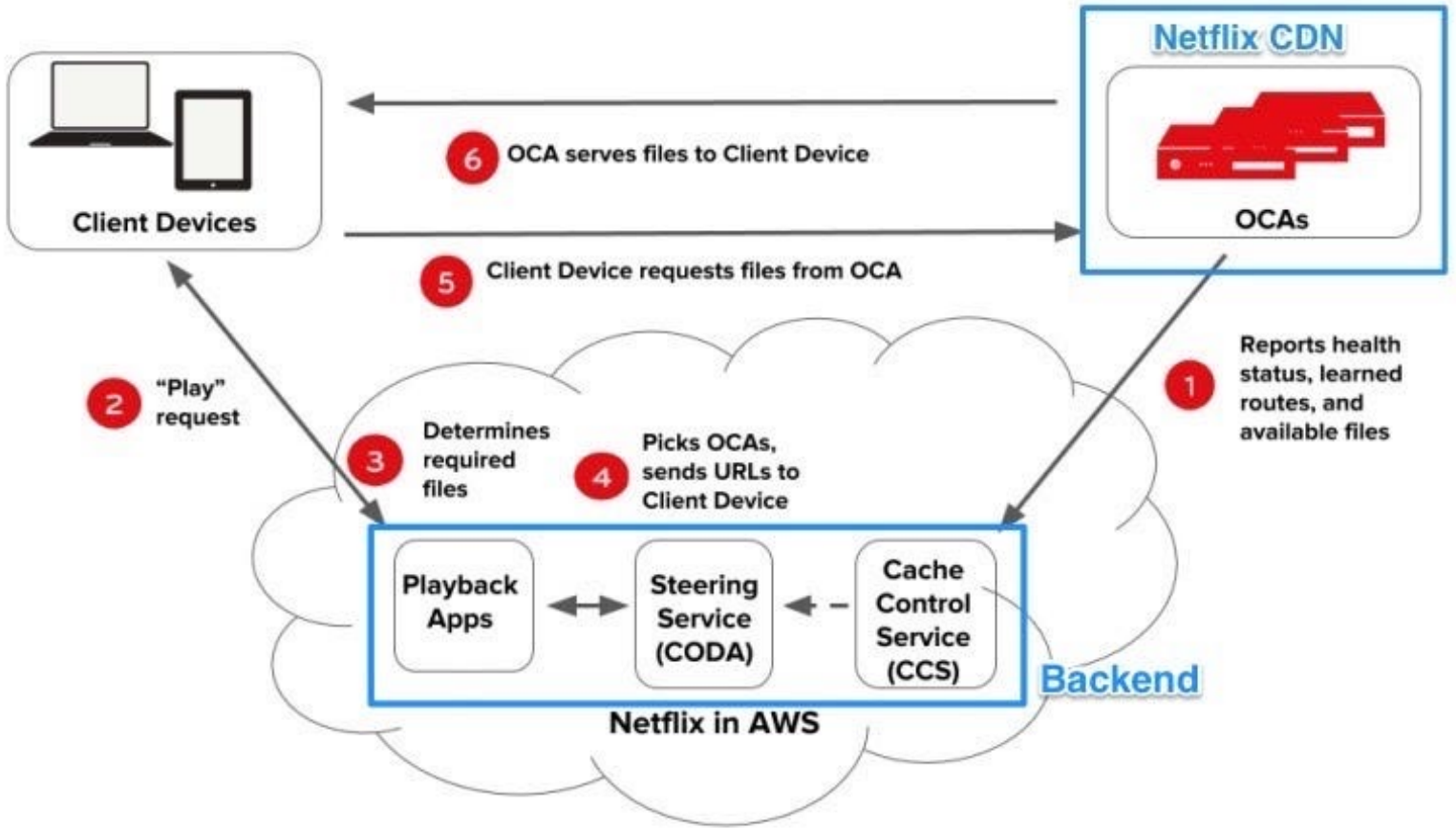
https://netflixtechblog.com/netflixs-viewing-data-how-we-know-where-you-are-in-house-of-cards-608dd61077da

Netflix billing migration to Cloud



https://netflixtechblog.com/netflix-billing-migration-to-aws-451fba085a4

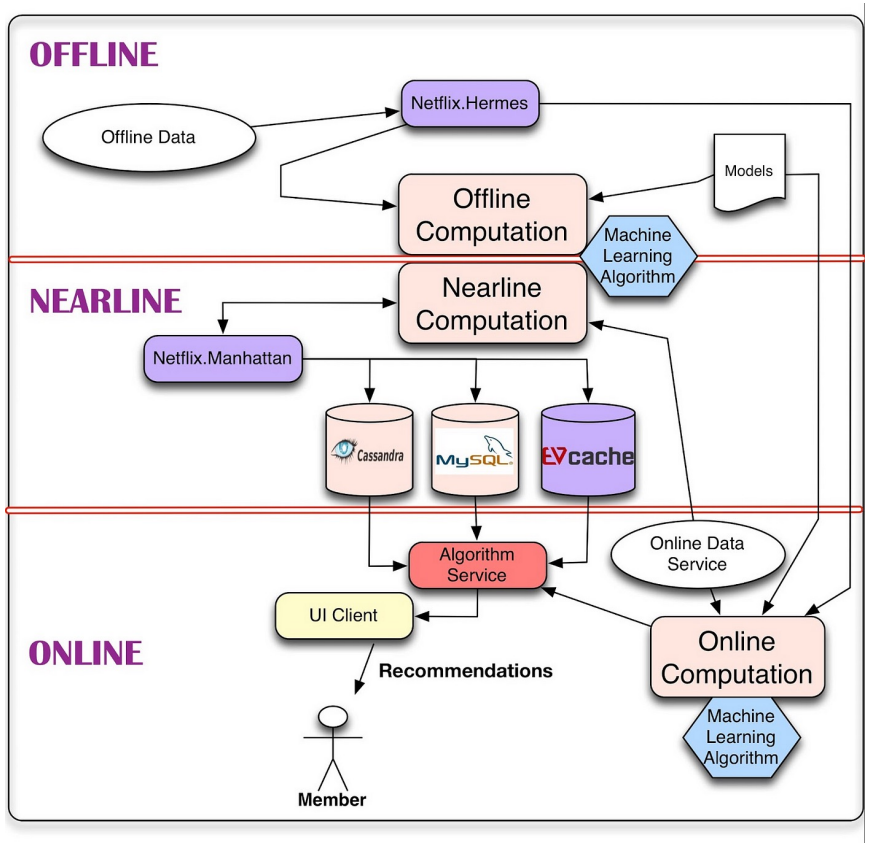
Netflix - Playback architecture for streaming videos

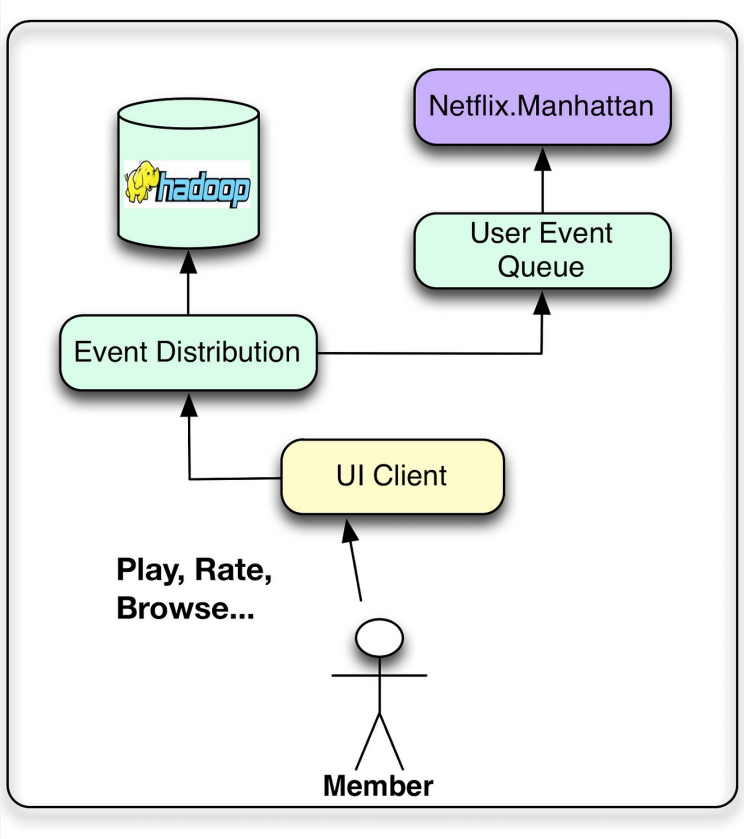


https://medium.com/swlh/a-design-analysis-of-cloud-based-microservices-architecture-at-netflix-98836b2da45f

Netflix recommendations

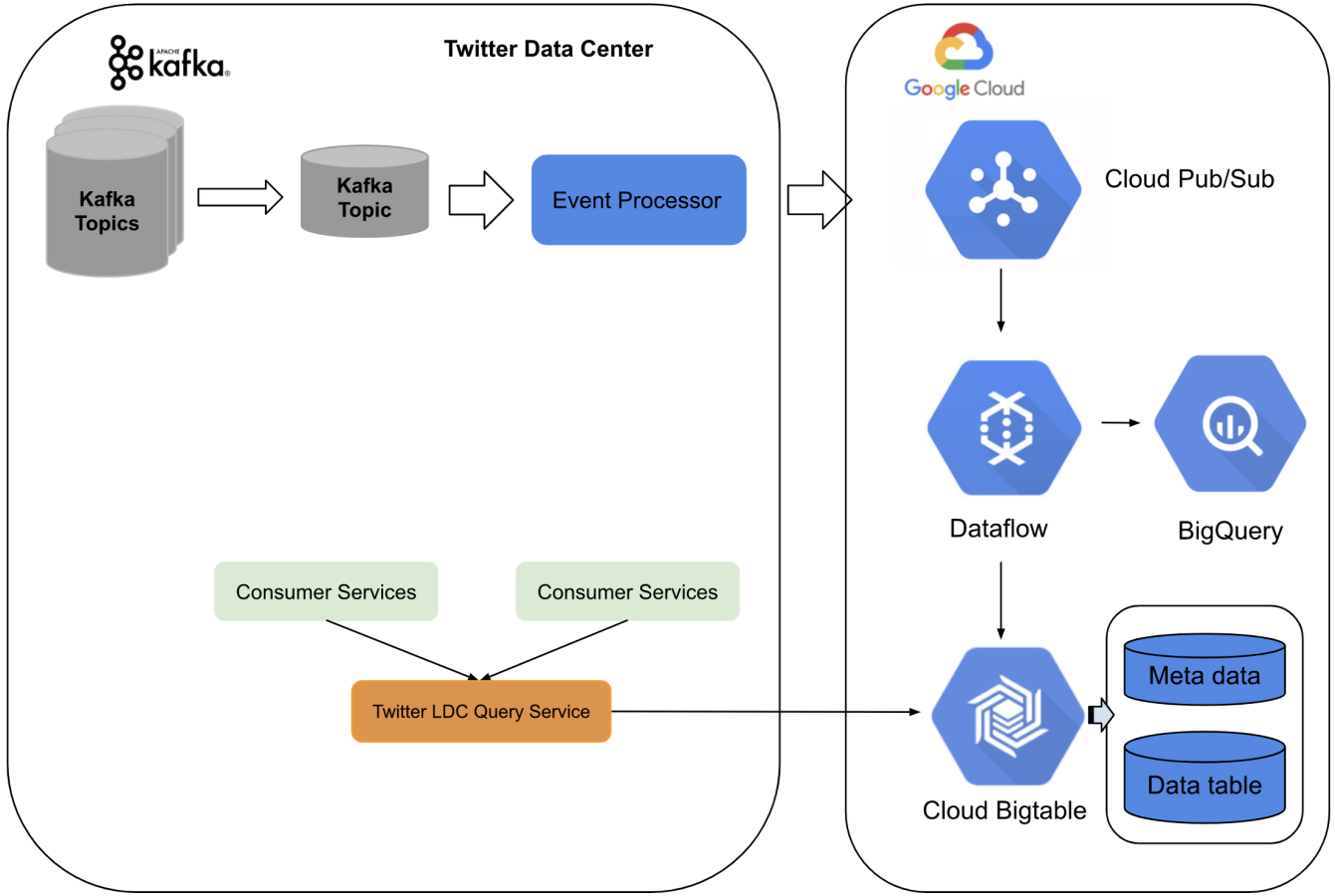
Digarams show control/data flow





https://netflixtechblog.com/system-architectures-for-personalization-and-recommendation-e081aa94b5d8

Twitter: Processing billions of tweets in real-time



https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/processing-billions-of-events-in-real-time-at-twitter -

Facebook: System architecture (Turbine)

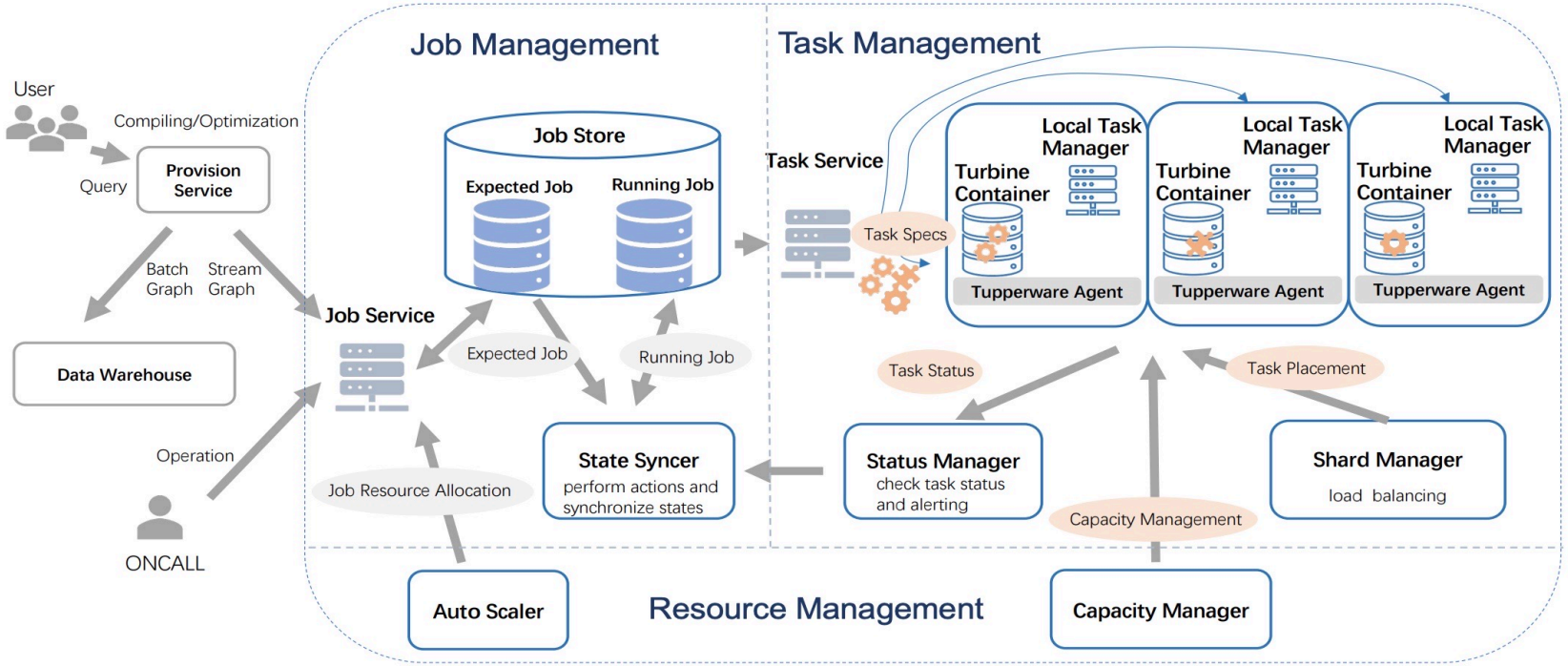


Fig. 2: Turbine's system architecture to manage stream processing services.

https://research.facebook.com/publications/turbine-facebooks-service-management-platform-for-stream-processing/

Facebook: Kangaroo - new flash cache optimized for tiny objects

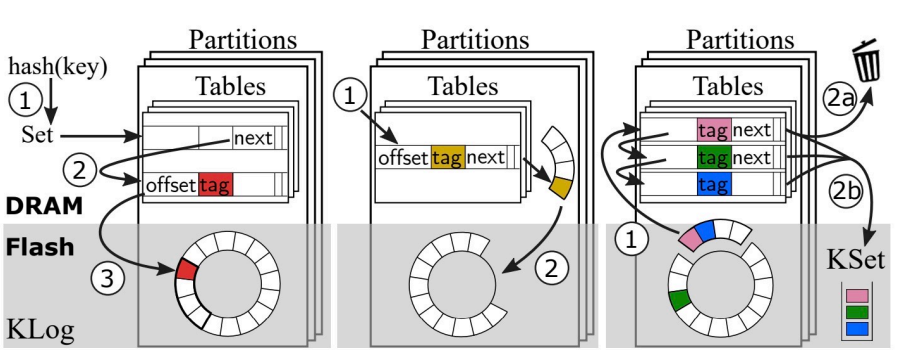
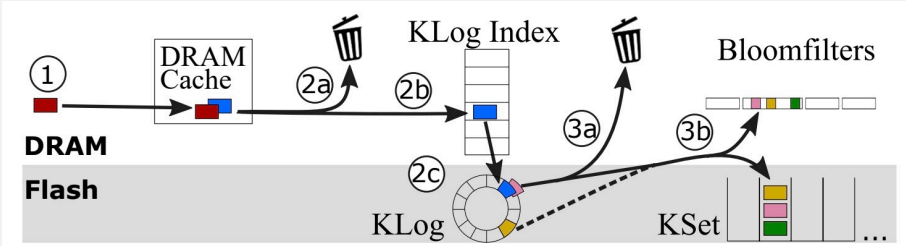
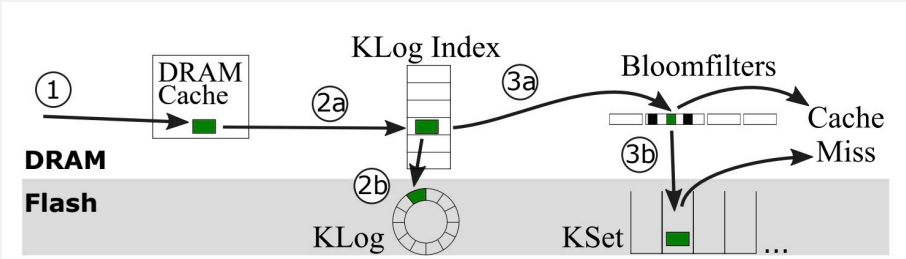


Fig. 4. Overview of KLog operations.



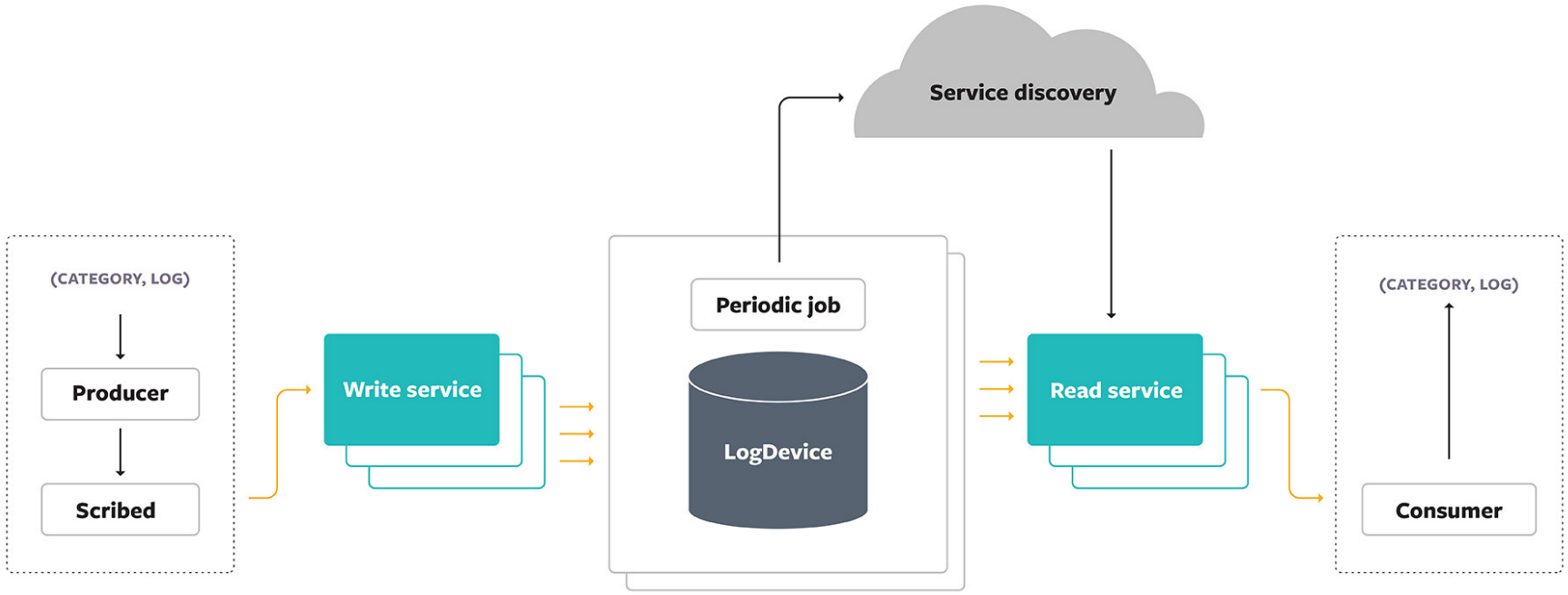
(b) Insertion and eviction in Kangaroo.



(a) A lookup in Kangaroo.

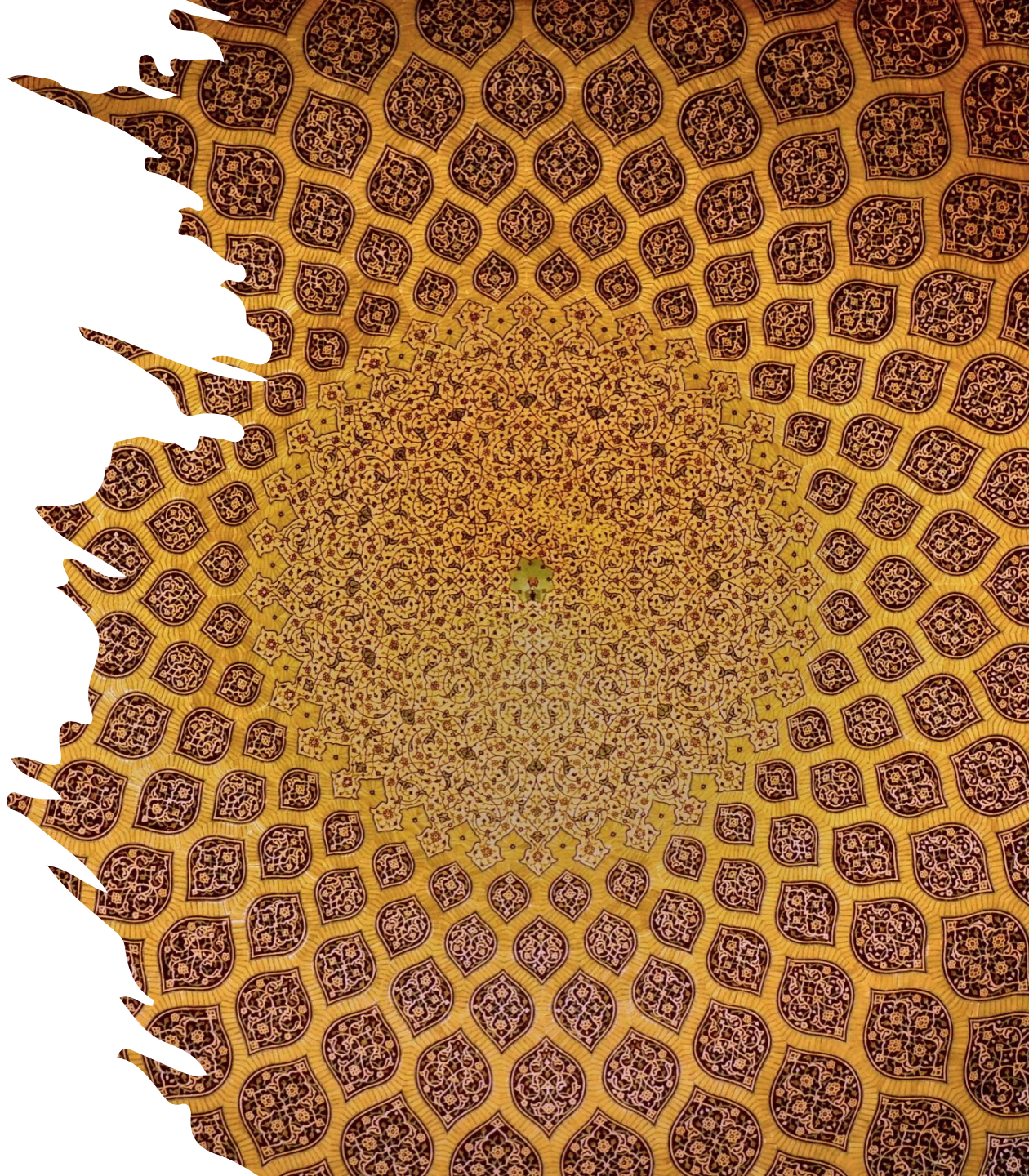
https://engineering.fb.com/2021/10/26/core-data/kangaroo/

Facebook: Scribe - a distributed, buffered queueing system



https://engineering.fb.com/2019/10/07/data-infrastructure/scribe/

Distinguishing Architecture, Model and Design



Architecture, Model, Design

Skeletal

Muscular

Facial

Structure vs. Behavior

Architecture

Structural components and their interconnections

Decisions that impact performance at scale

Model

Behavioral components and their interactions

Decisions that impact the system behavior

Models can be developed for multiple purposes

Model for understanding

Model for analysis (formal verification and validation etc.)

Model for simulation (validation by running through scenarios)

Model for engineering (design, code generation, testing)

What should be modelled?

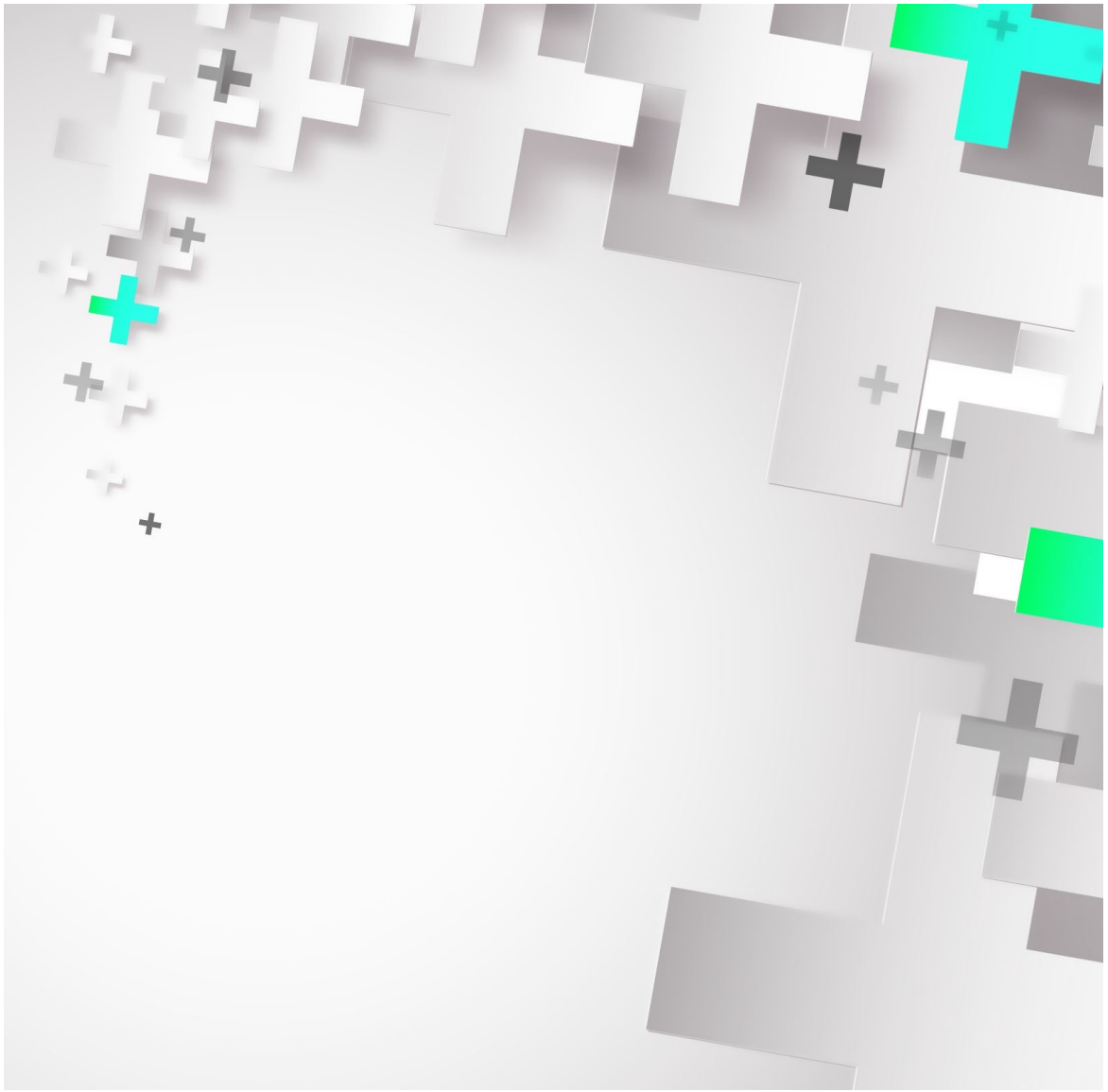
Structure and components

Behaviors and interactions

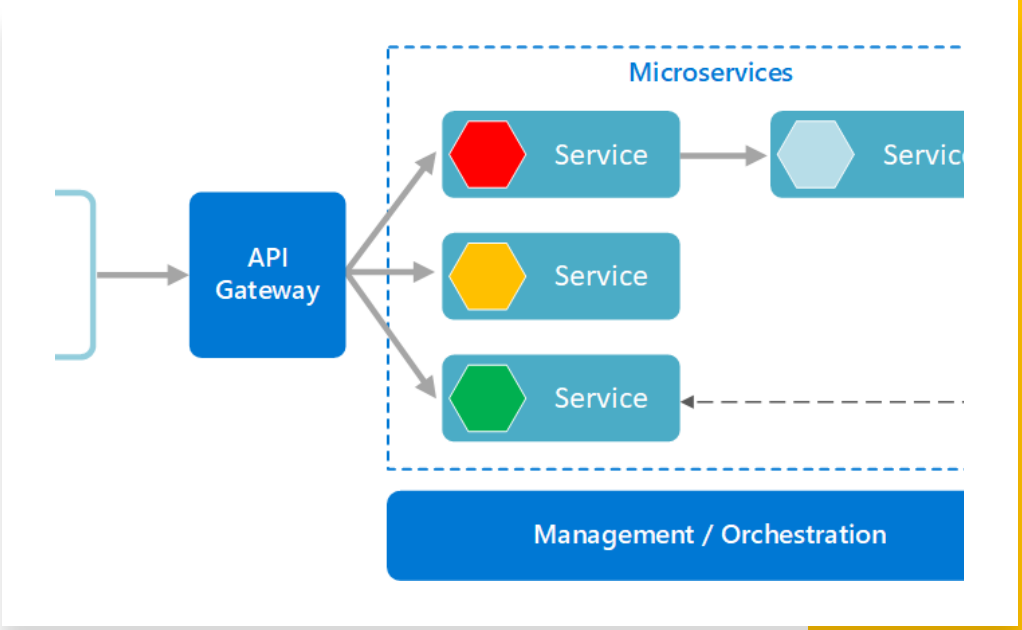
Data

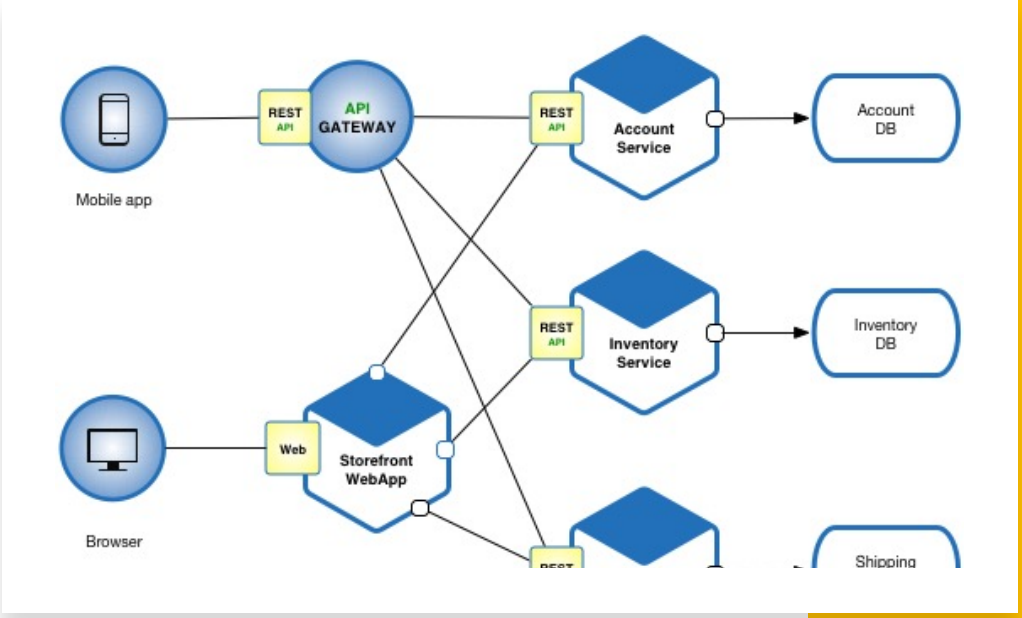
Context

Our focus is application software



Application software are system of systems (SoS)





We will model these system of systems (SoS) in this course

Requirements (Behavioral model)	Structure (Structural model)	Interactions (Interaction model)
• Specify expected behavior of SoS in terms of use cases or input-output examples	• Specify structure of the SoS in terms of component systems and their interconnections	• Specify the interactions and rules of interactions for component systems as well as the SoS

Operating at an abstraction level that provides understanding of the systems involved to the desired degree

Written models are desirable

Unambiguous specification

Clear communication

General-purpose models: UML (including SysML, Statecharts)

Electronics Design models: Verilog, VHDL

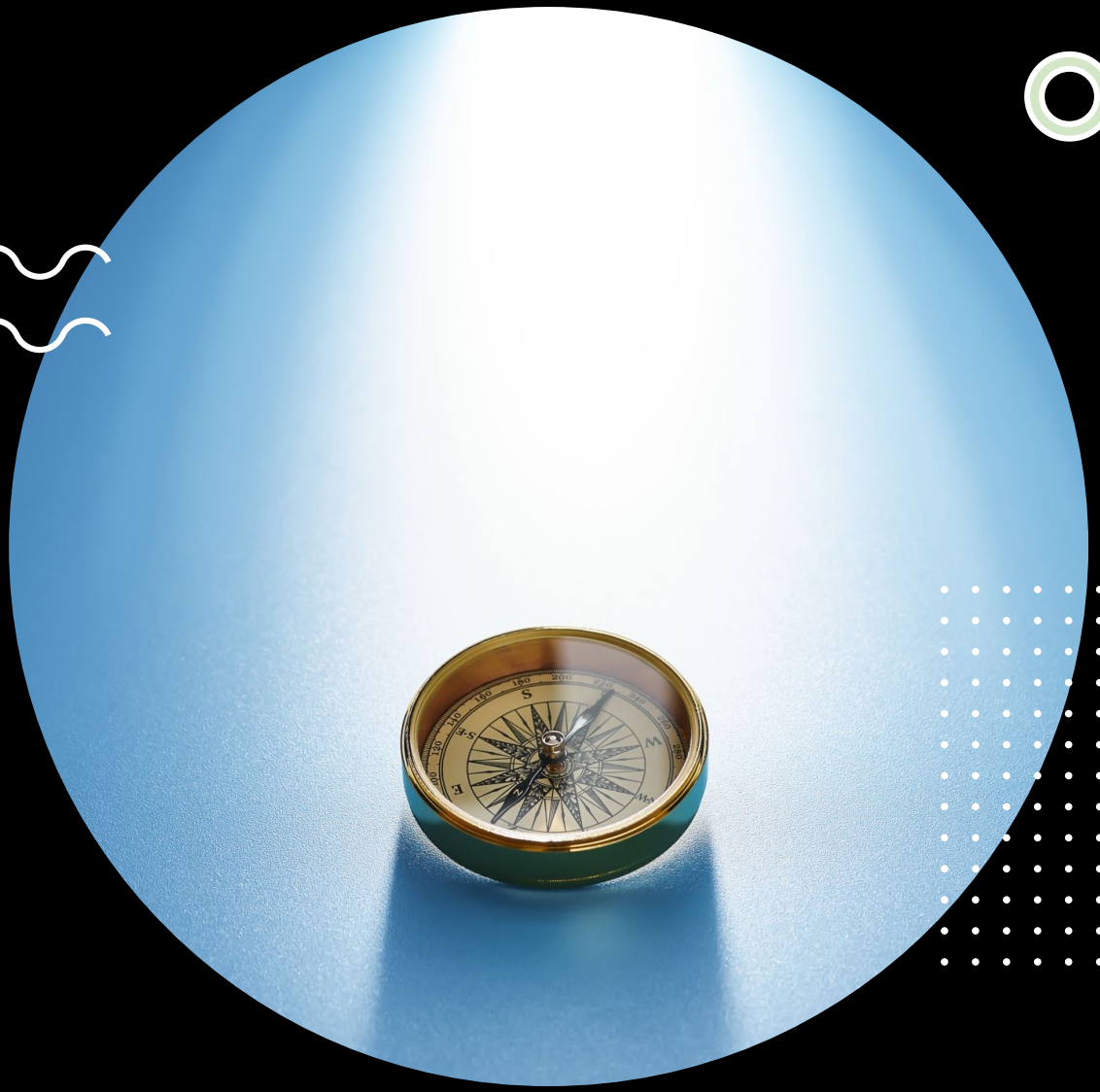
Dynamic systems models: Simulink

Models for formal analysis: PVS, UPPAAL, PRISM, Promela

Many domain-specific languages, including ADLs (AADL, ADDA)

Throughout this unit, we will explore the idea of Transition Systems formalism as a Modeling Language (TSML)

Questions?



Class exercise

Read the article about one of the Twitter features (Ad pacing)

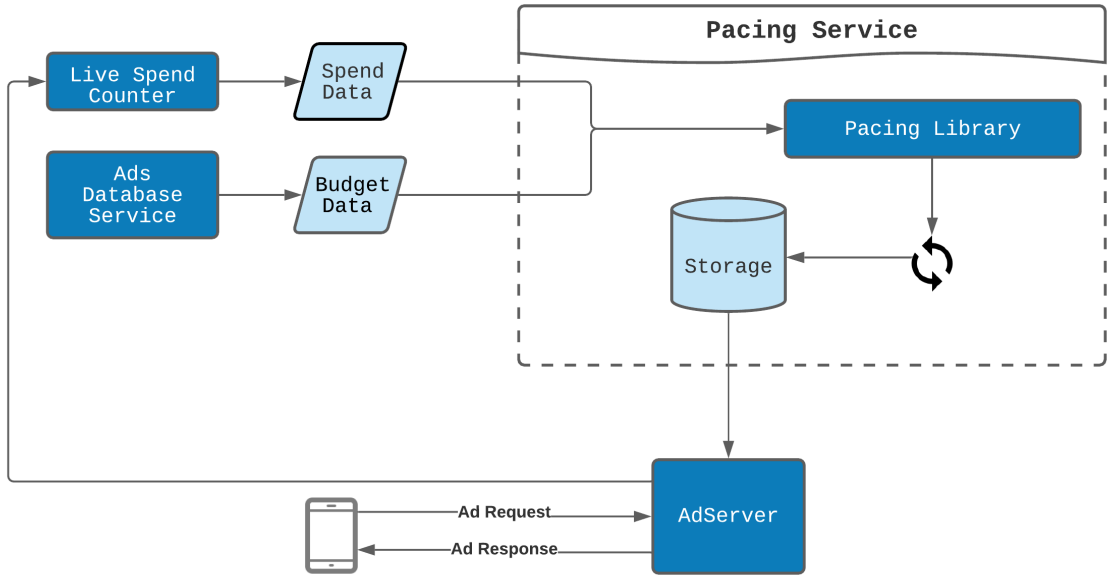
Identify these parts of the blog: Architecture description, Model description, Design description

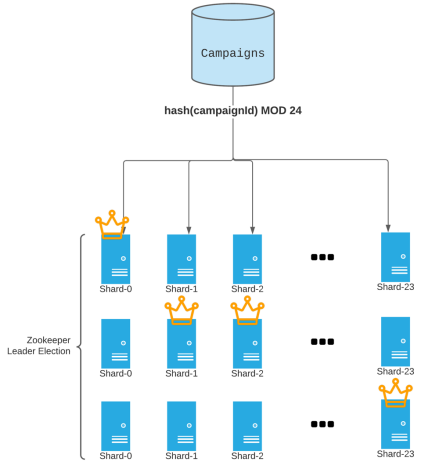
Summarize the article in 300-350 words, under three sections: Architecture, Model and Design

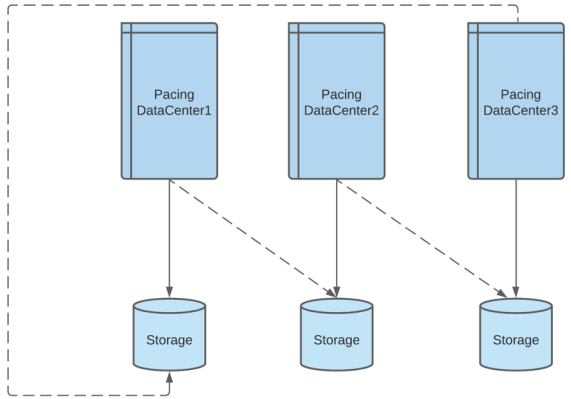
Write the ad pacing system as 6-tuple of a transition system.

Email your summary and model to: ganne.jyotheeswar@students.iiit.ac.in, with subject: Unit3-Session2-Classwork-Team<id>

Twitter: Pacing the ads







https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/how-we-built-twitter-s-highly-reliable-ads-pacing-service

Twitter

- Leader election
- Cross DC replication

https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/how-we-built-twitter-s-highly-reliable-ads-pacing-service

Questions?