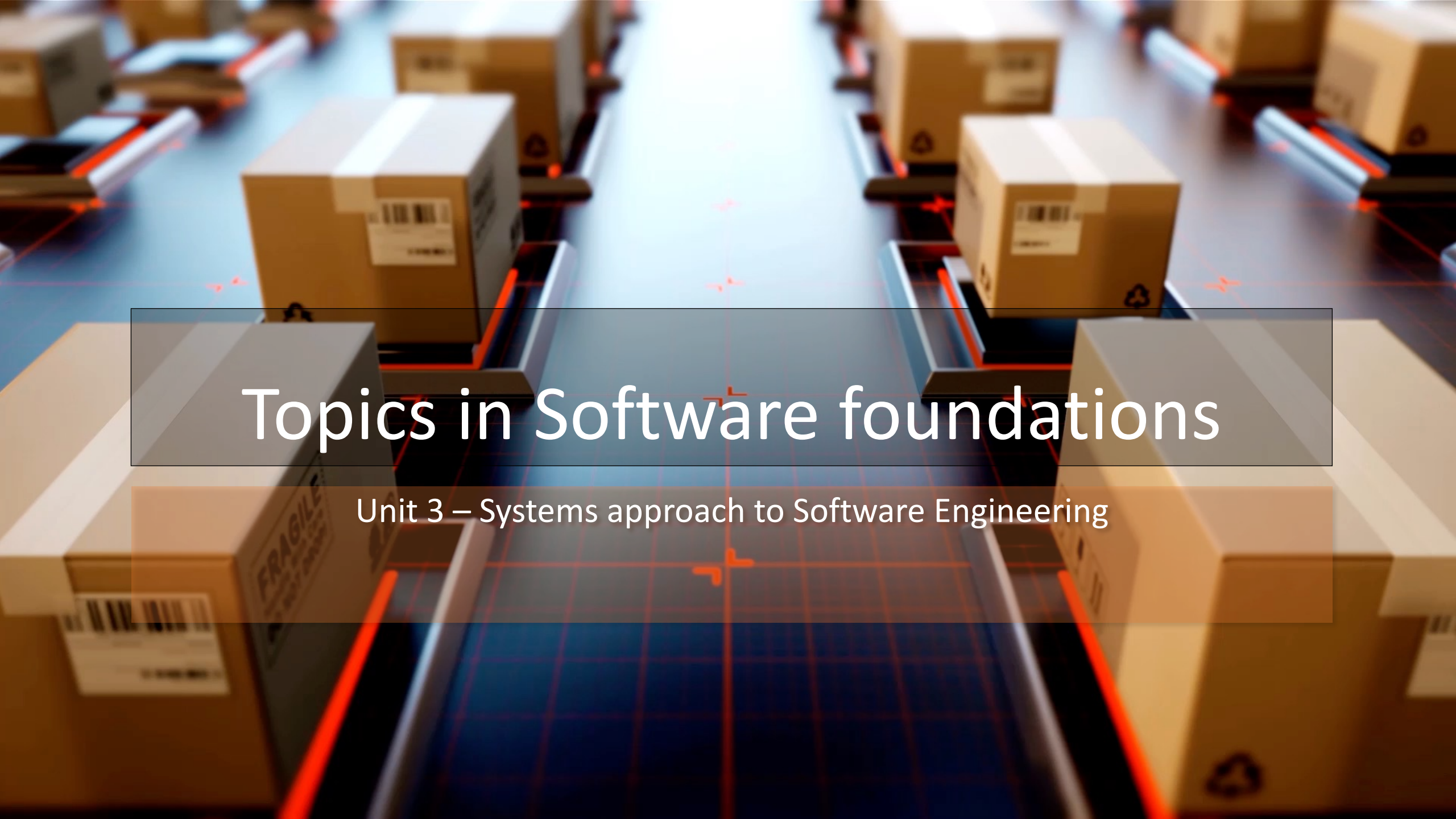


Topics in Software foundations

Unit 3 – Systems approach to Software Engineering

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

Recap/continuation from the last class

System modifications for various types of problems

Richness of UI	Quality of client-server connection	Quality of client device	Patterns of application access

• There may be multiple data sources that compose a page
  • Multiple MVC systems at play, web page view is composed of multiple of these Views.
• A page component may include data from multiple sources
  • A new data model is created to map to the required view, MVC using this model is used.
	• Low bandwidth
	  • Reduce functionality (impacts U), reduce user elements (impacts Y)
	• High latency
	  • Take local decisions on client – impacts transition functions. Requires sync with server for decisions, other issues
	• Frequent connection drops
	  • Reduce connection setup time, apply high latency solution
		• Low processing power (low-end android phone)
		  • Lite version (functionally)
		• Low display capabilities (watch)
		  • Lite version (visually)
			• The user may use multiple devices to access the system simultaneously
			  • As long as observer is in place and in use, this should work. Concurrent access needs to be available.
			• A large number of users access the system simultaneously
			  • Gateway to route the traffic - split into batches which can be handled by existing system
			  • Refine the solution to have better scale characteristics

System modifications for various types of problems

There may be multiple data sources that compose a page

- Multiple MVC systems at play, web page view is composed of multiple of these Views.

A page component may include data from multiple sources

- A new data model is created to map to the required view, MVC using this model is used.

Low bandwidth

- Reduce functionality (impacts U), reduce user elements (impacts Y)

High latency

- Take local decisions on client (impacts F). Requires sync with server for decisions made locally
- Reduce message chatter with server

Frequent connection drops

- Reduce connection setup time, apply high latency solution

Low processing power (low-end android phone)

- Lite version (functionally)

Low display capabilities (watch)

- Lite version (visually)

The user may use multiple devices to access the system simultaneously

- As long as observer is in place and in use, this should work. Concurrent access needs to be available.

A large number of users access the system simultaneously

- Gateway to route the traffic - split into batches which can be handled by existing system
- Refine the solution to have better scale characteristics

Modeling Data Access

Unit 3 – TiSF S'23

Session 6 (2023-04-17)

Two categories of data

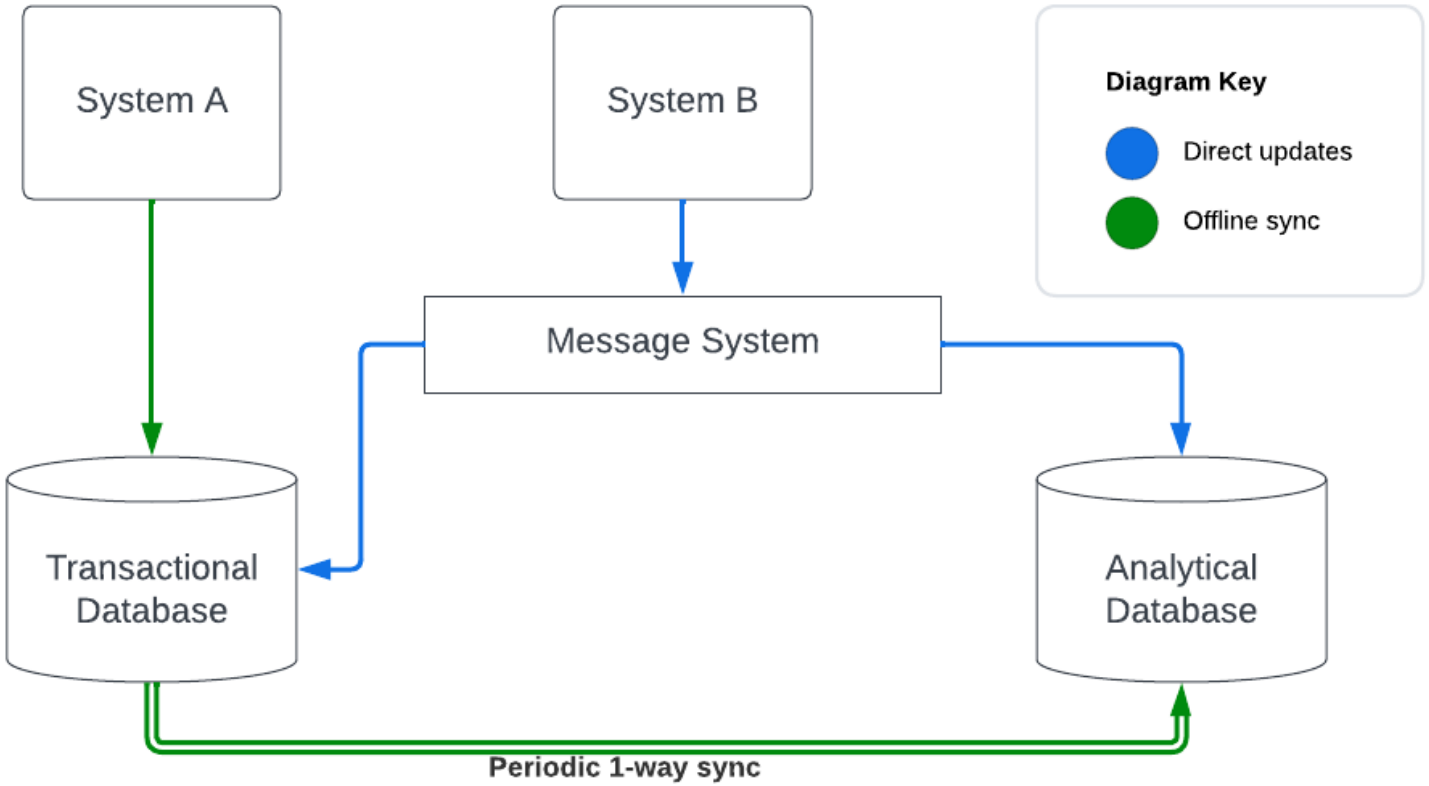
Transactional

- State data for a system (and SoS)
  - Appointments booked by this user this year
  - Money collected in every transaction in a salon since beginning of time
- Action (U) and Observable (Y) may have data associated with them
  - Booking required for a specific user for a particular date and time
  - All empty calendar slots in response to a query about availability

Analytical

- Record for later use (Archive)
  - Log every system communication message
  - Audit trails of all payment collection
- Analysis and insights
  - Data warehouse for business performance reporting
  - Demand forecast system using transaction data

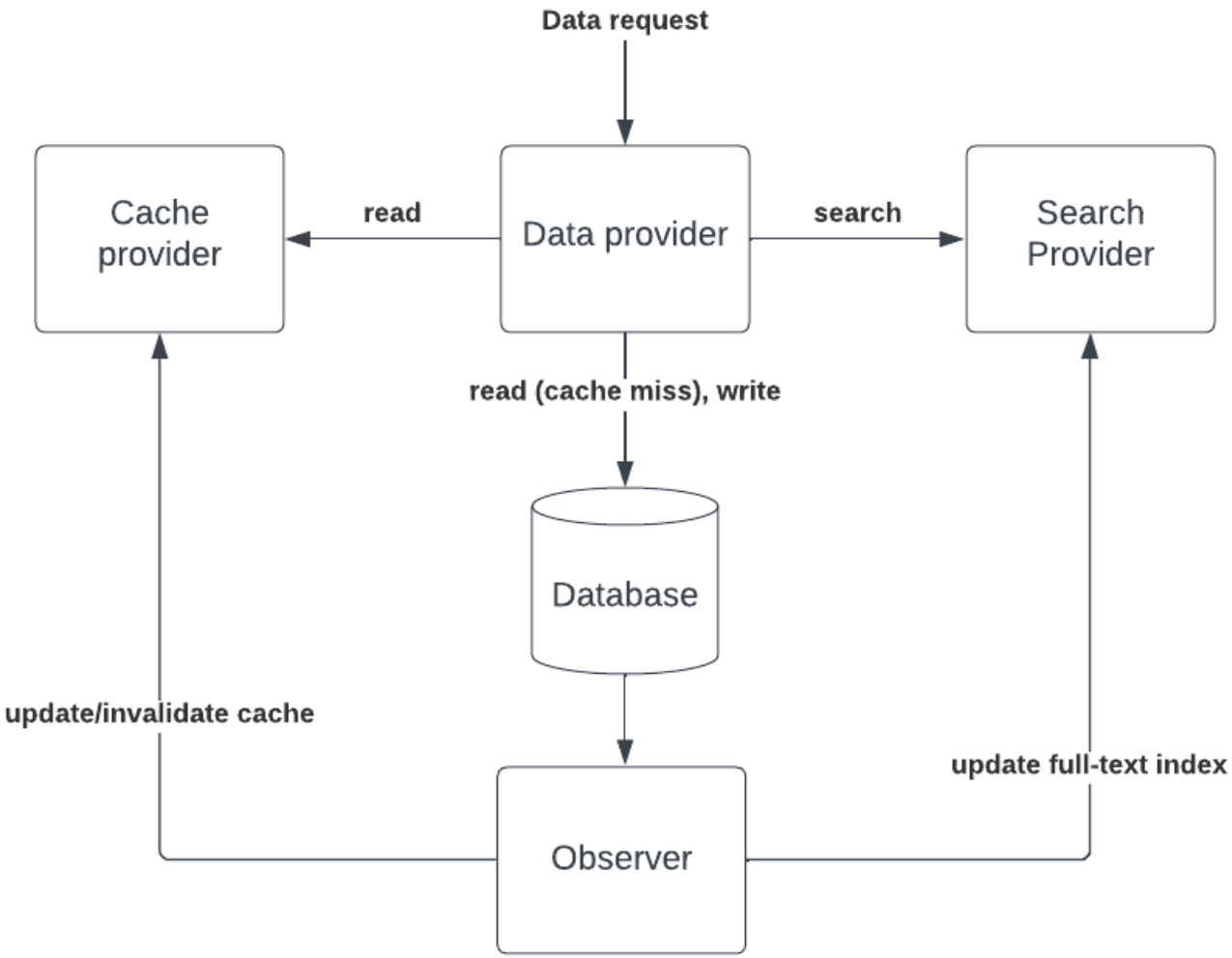
Transaction and analytical data



Various ways data can be held by the system

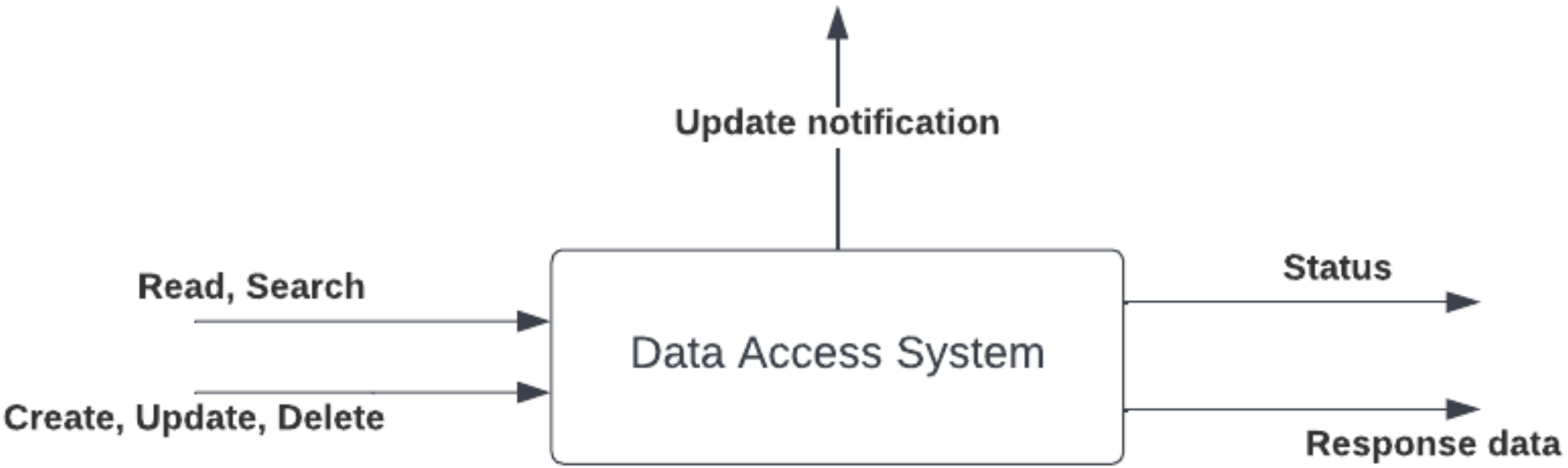
<table>
<tr><td colspan="4">Data Access System</td></tr>
<tr><td>Internal Service API</td><td>External Service API</td><td>Querying language API</td><td>File API</td></tr>
<tr><td>Internal data service</td><td>External data service</td><td>Database system</td><td>File System</td></tr>
<tr><td colspan="4">Operating System Services (Network, Storage)</td></tr>
</table>

Typical data access system model

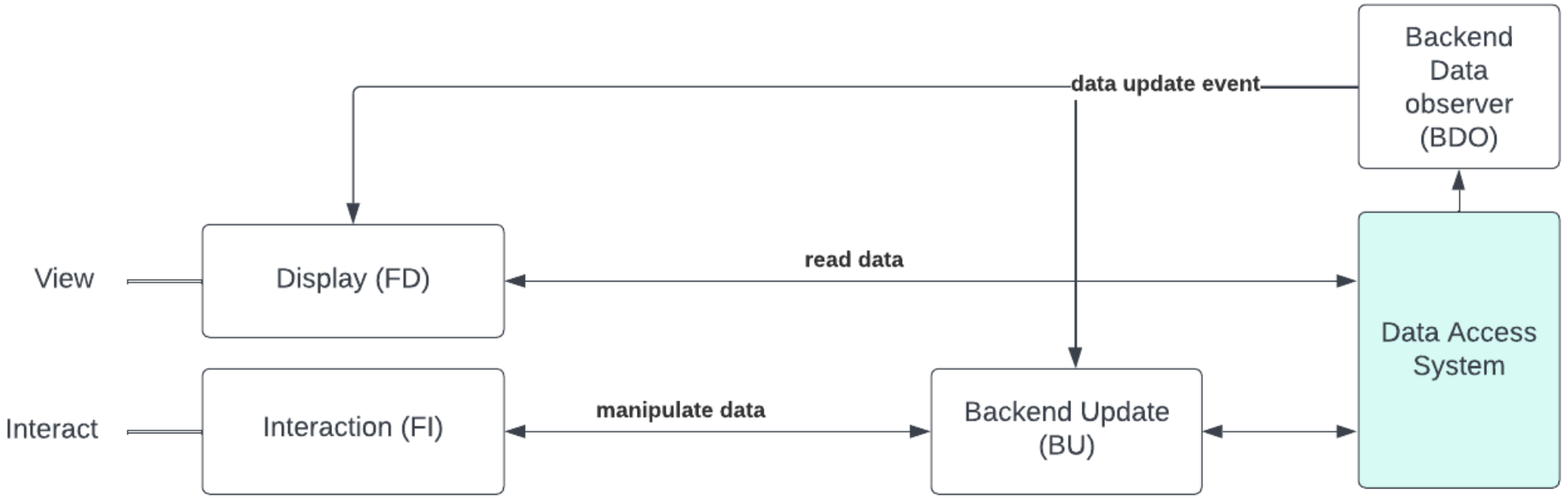


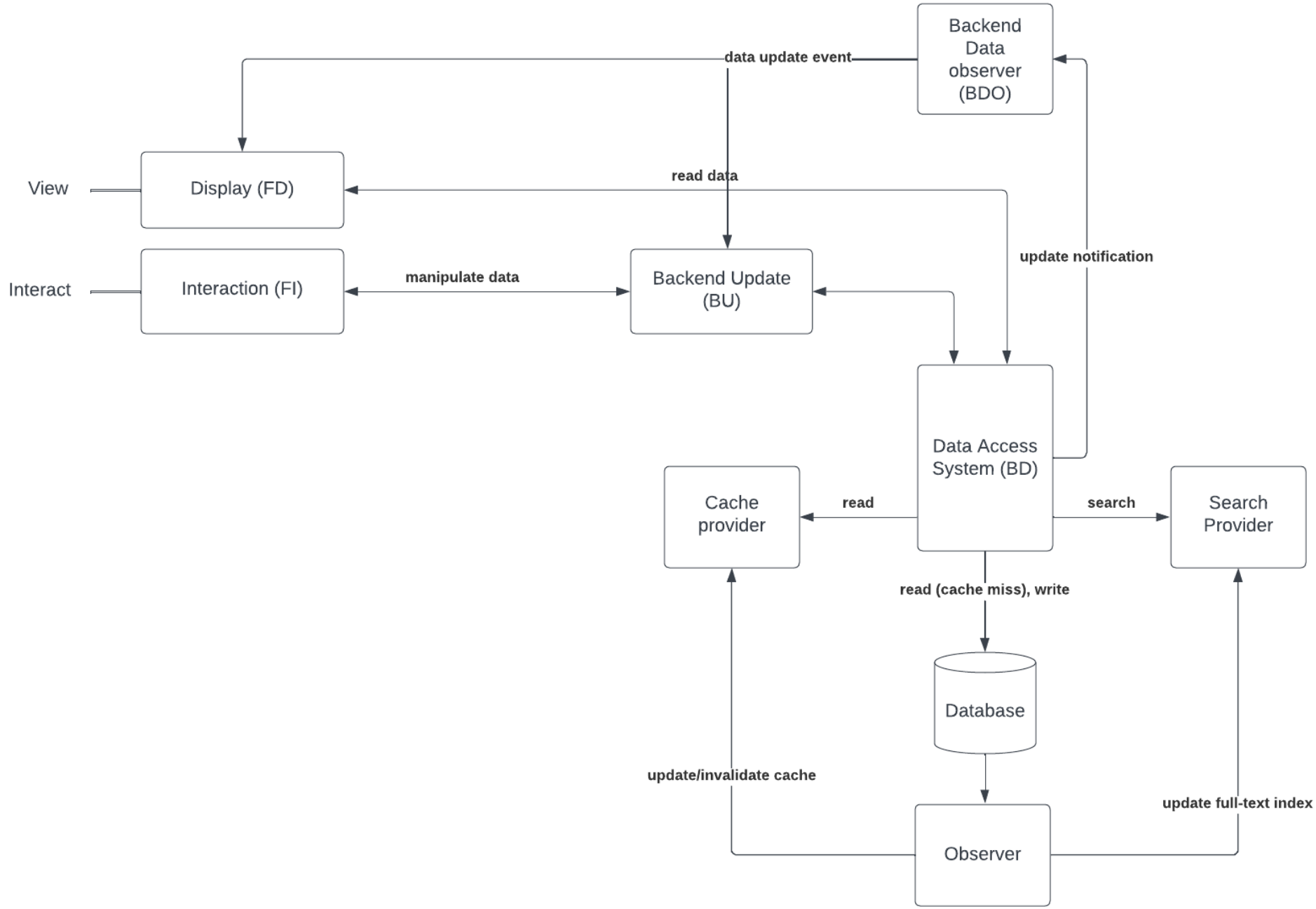
Designing data-intensive applications

Data access system



UI System (discussed in last class)




Backend Data observer (BDO)
data update event
View
Display (FD)
read data
Interact
Interaction (FI)
manipulate data
Backend Update (BU)
update notification
Data Access System (BD)
Cache provider
read
search
Search Provider
read (cache miss), write
Database
update/invalidate cache
update full-text index
Observer

Data scenarios that will require model tweaks

A large amount of data in every read response
Too many data access requests to render a UI
Many more reads compared to writes
Database server going down
Too much data for a single database server

To be cont*inued in next class..*