Distributed Systems
A finer perspective

Agenda

- About Me
- Fallacies of Distributed System
- Ordering
- Ordering in Distributed systems
- Current Solutions : Drawbacks
- Why does Ordering matter ?
- Eventing : Eventing and Event Types
- Guaranteeing Ordering
- Functional Decomposition Pitfalls - Structural Architecture
- Separation of Concerns - Structural Architecture
- Quotes To Live By
- References

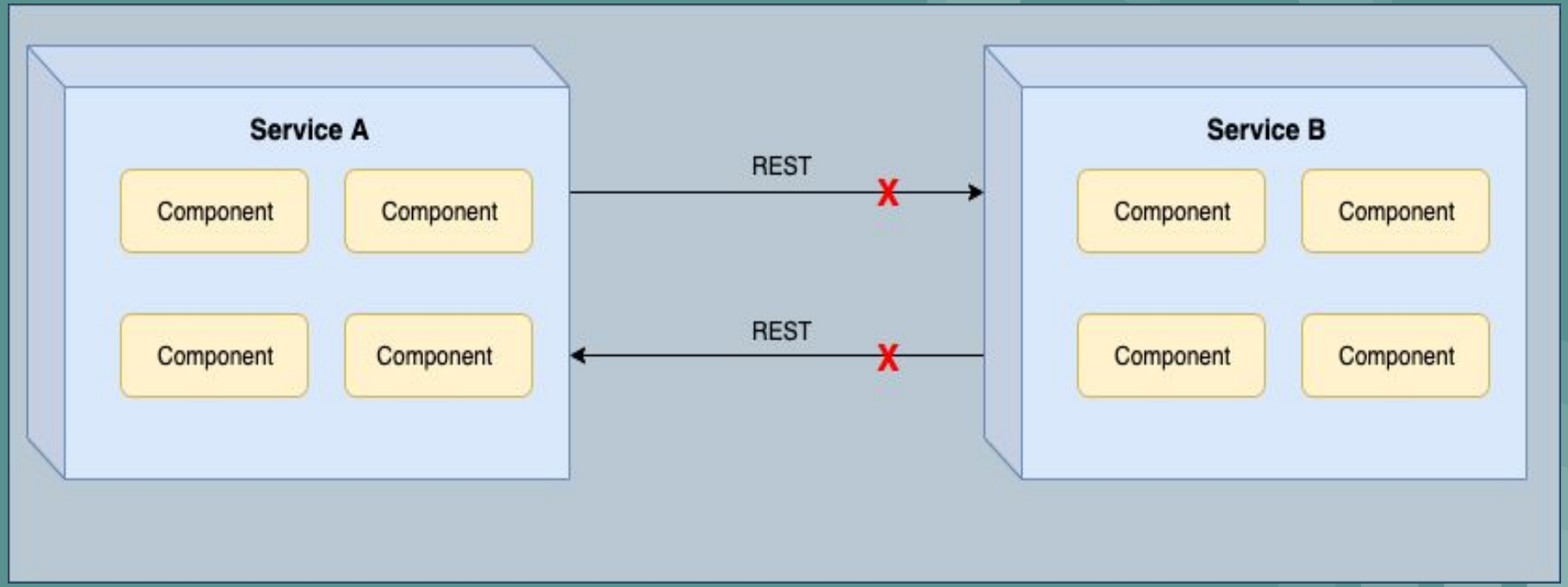
About Me

- Software Developer with two decades of experience
- Polyglot - C++, Java, Erlang, Haskell, Scala, Closure, Golang
- Author - Go The Other Way - A book on golang
- Interested in Distributed Systems and Architecture
- Aspiring Speaker and Developer Advocate
- Former National Level Tennis Player and former Ranked Senior ITF player

http://rethingkingdesign.tech
https://linkedin.com/in/maneeshc

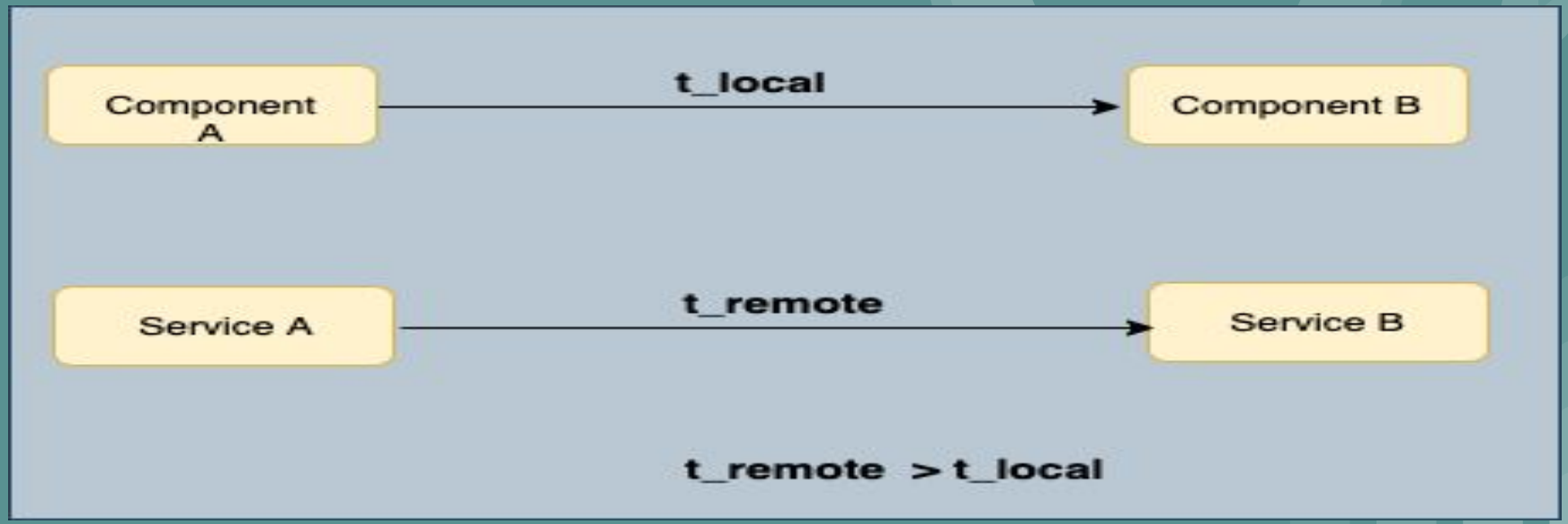
Fallacies of Distributed Systems

- The Network is reliable.



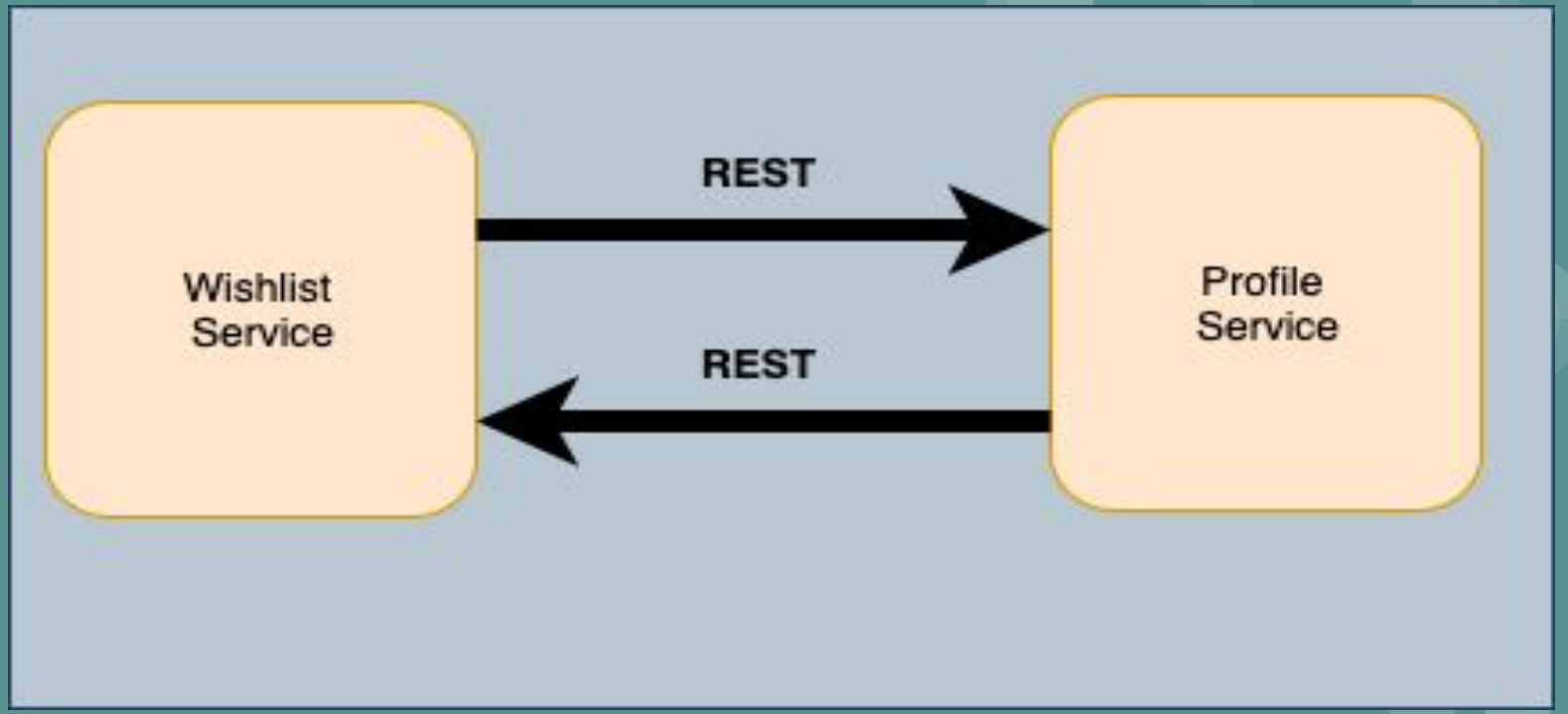
Fallacies of Distributed Systems

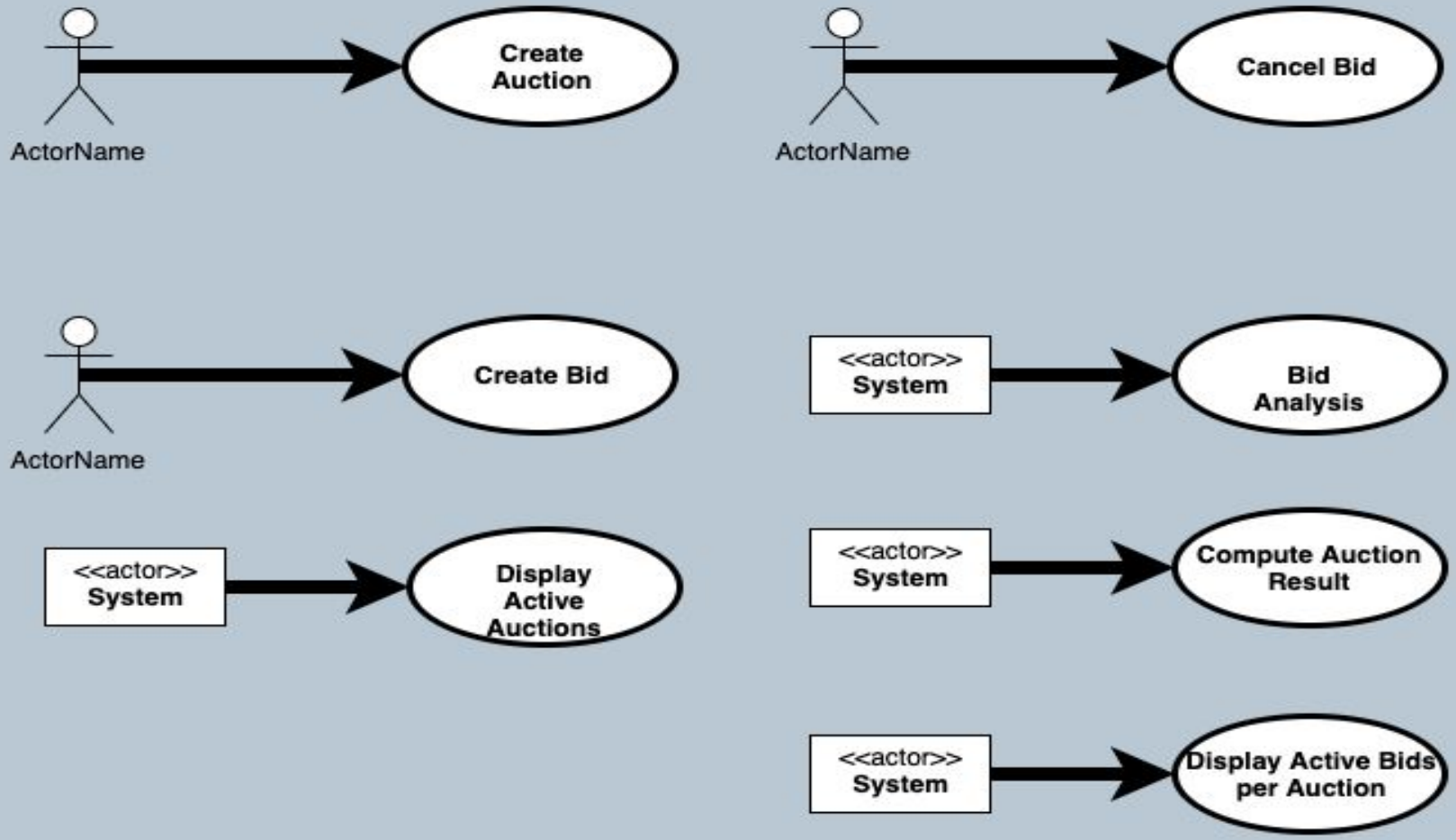
- Latency is Zero.

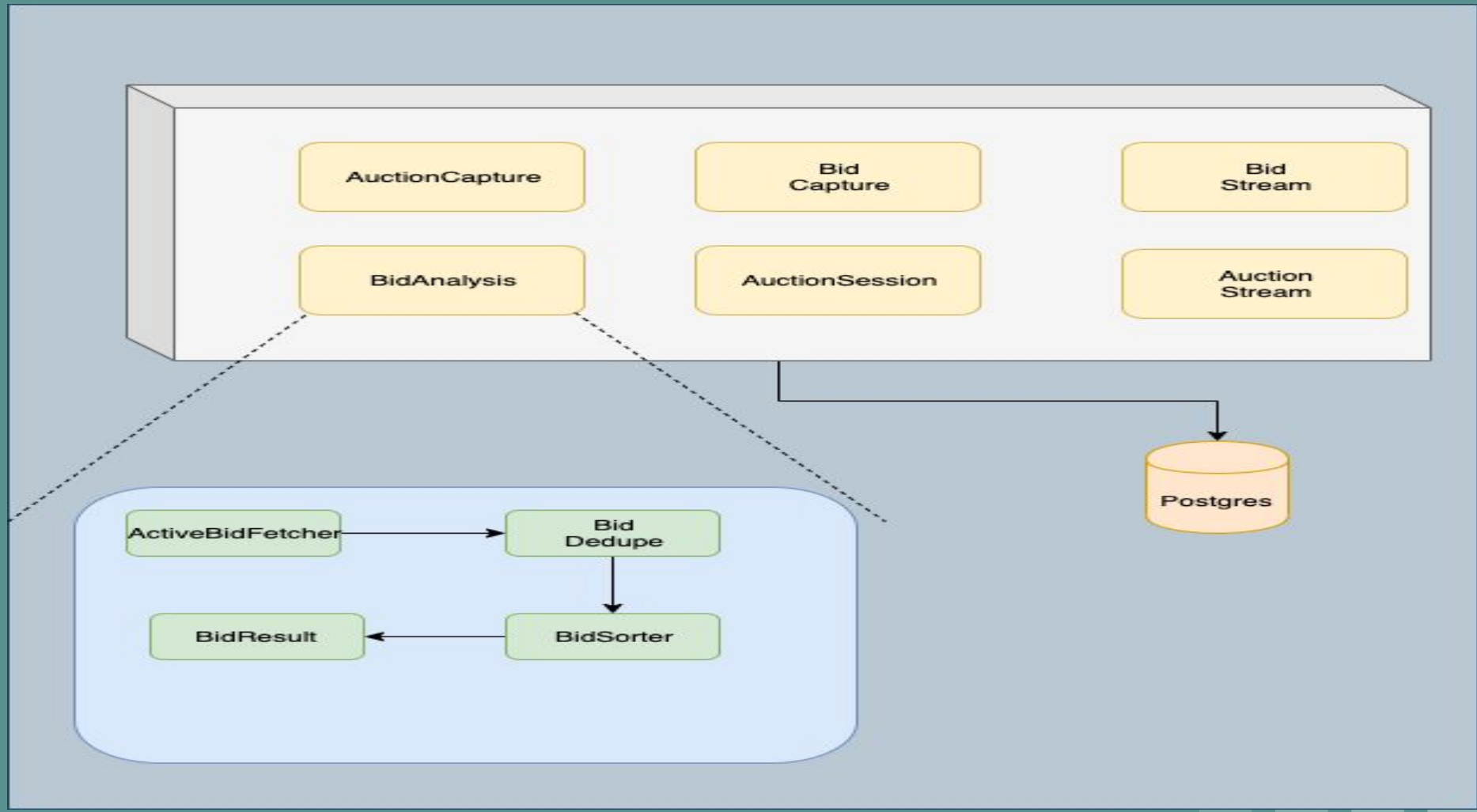


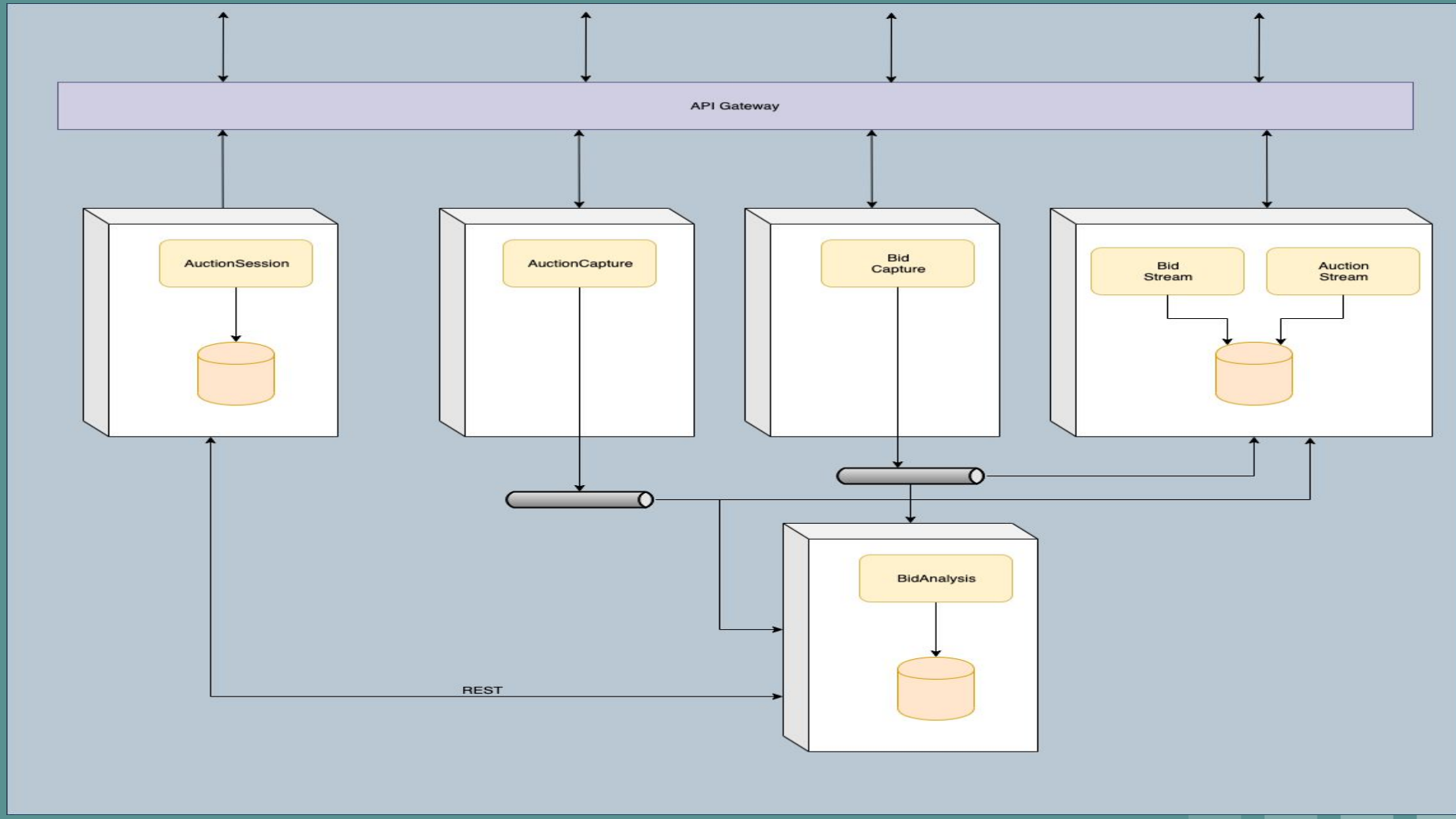
Fallacies of Distributed Systems

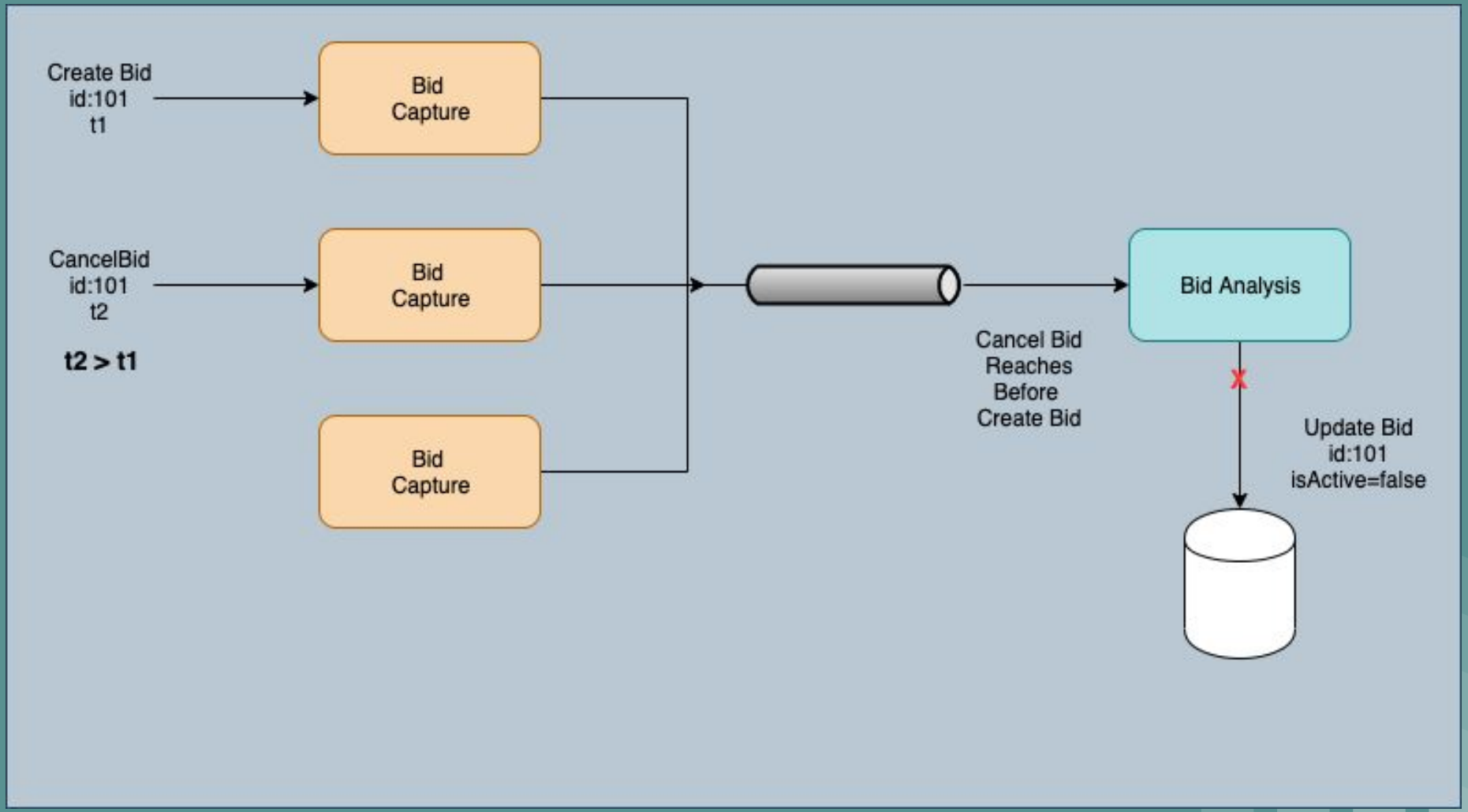
- Bandwidth is infinite

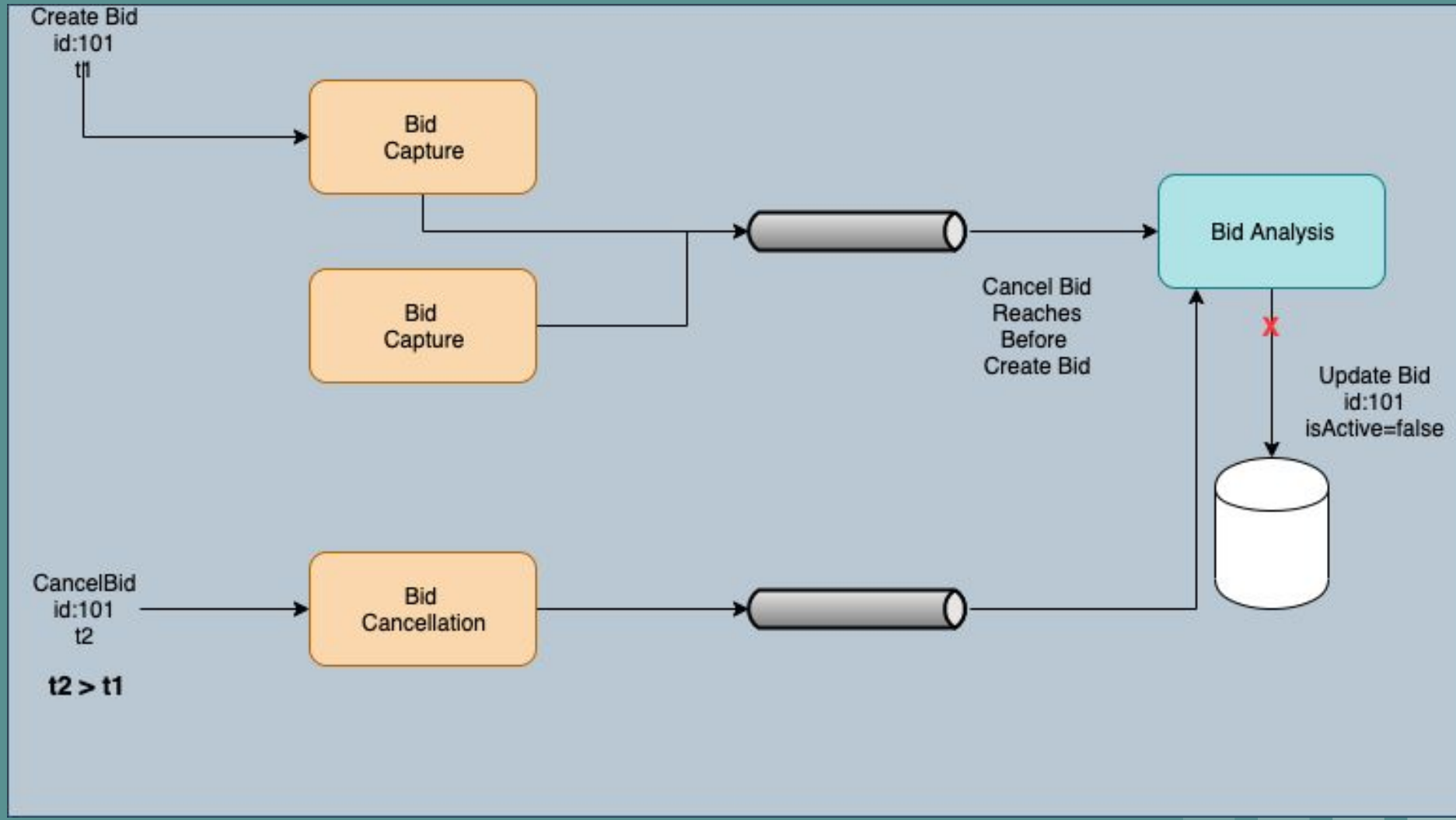



ActorName
Create Auction
ActorName
Cancel Bid
ActorName
Create Bid
<<actor>> System
Bid Analysis
<<actor>> System
Display Active Auctions
<<actor>> System
Compute Auction Result
<<actor>> System
Display Active Bids per Auction


AuctionCapture
Bid Capture
Bid Stream
BidAnalysis
AuctionSession
Auction Stream
Postgres
ActiveBidFetcher
Bid Dedupe
BidResult
BidSorter

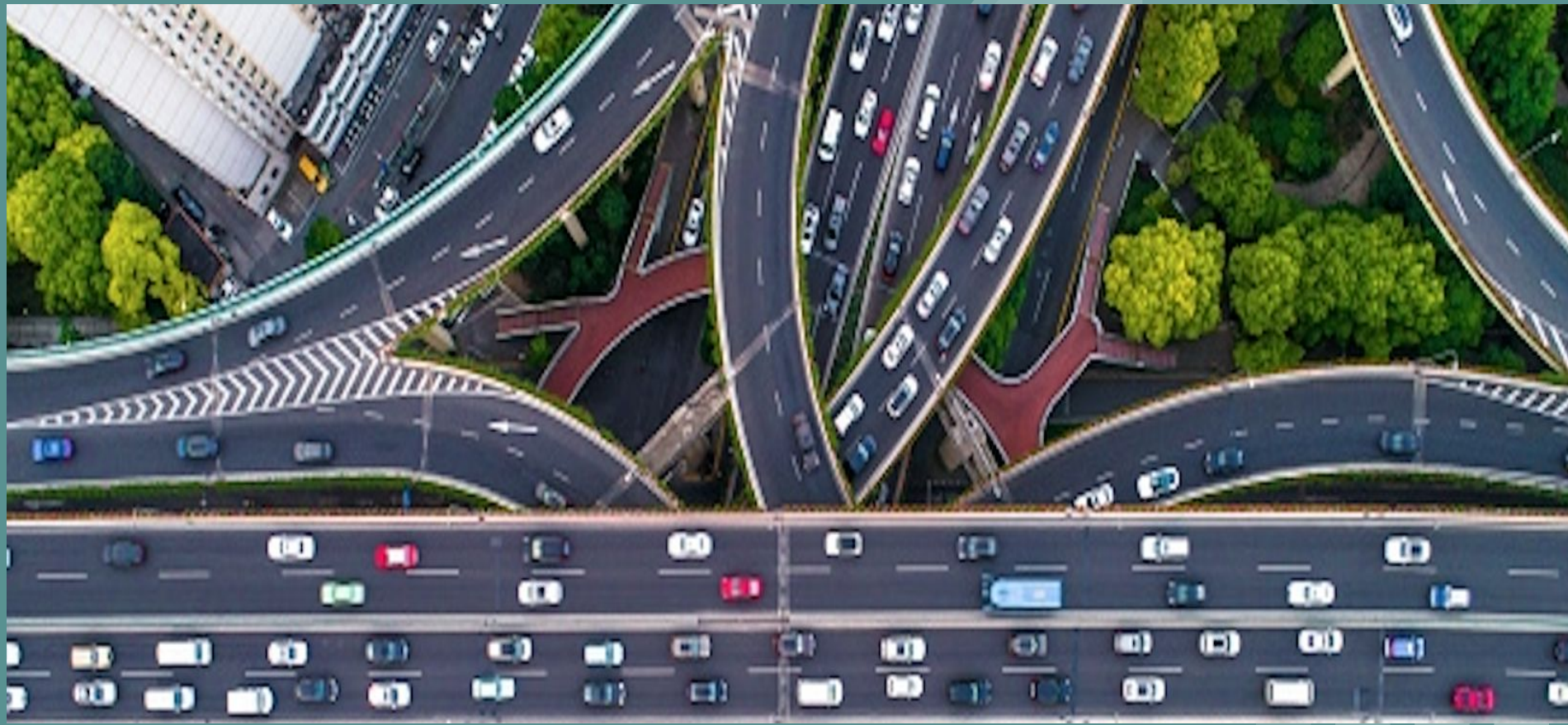

API Gateway
AuctionSession
AuctionCapture
Bid
Capture
Bid
Stream
Auction
Stream
BidAnalysis
REST


Create Bid
id:101
t1
Bid
Capture
CancelBid
id:101
t2
t2 > t1
Bid
Capture
Bid
Capture
Cancel Bid
Reaches
Before
Create Bid
Bid Analysis
Update Bid
id:101
isActive=false

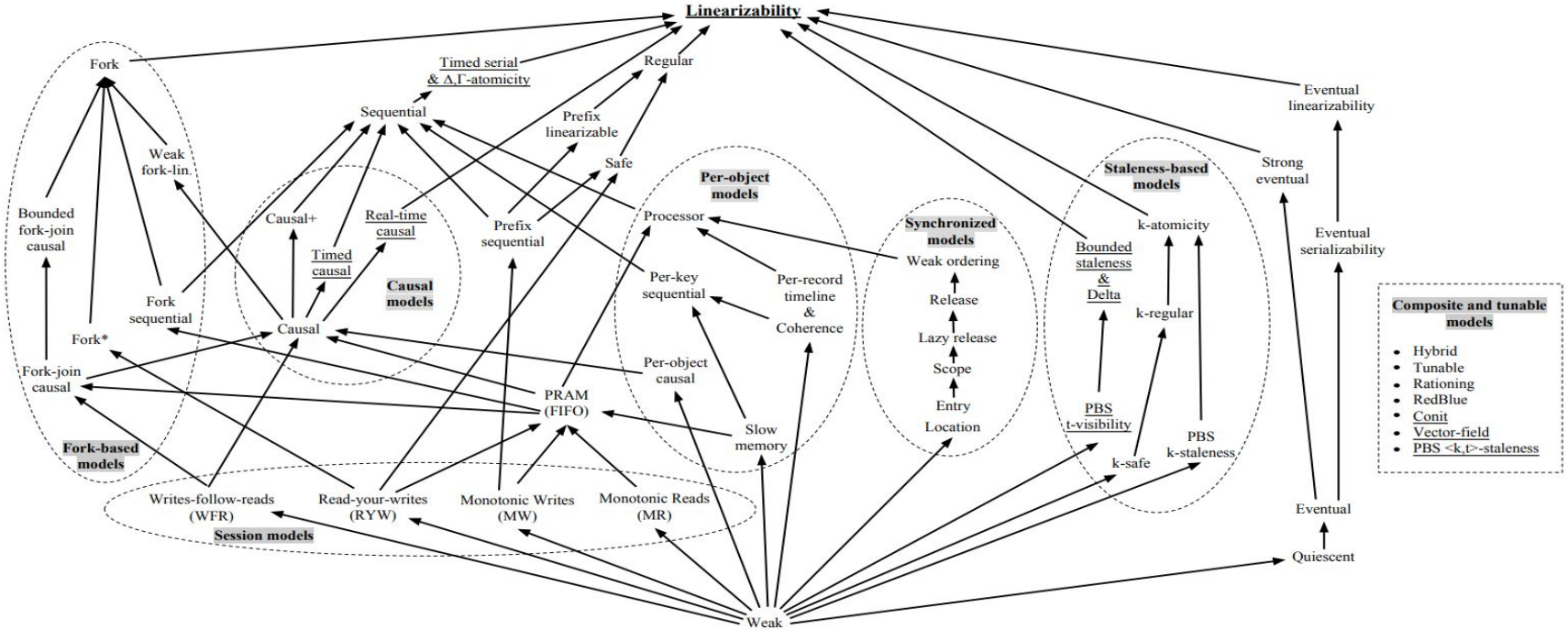

Create Bid
id:101
t1
Bid
Capture
Bid
Capture
Bid Analysis
Cancel Bid
Reaches
Before
Create Bid
X
Update Bid
id:101
isActive=false
CancelBid
id:101
t2
Bid
Cancellation
t2 > t1

Total Order

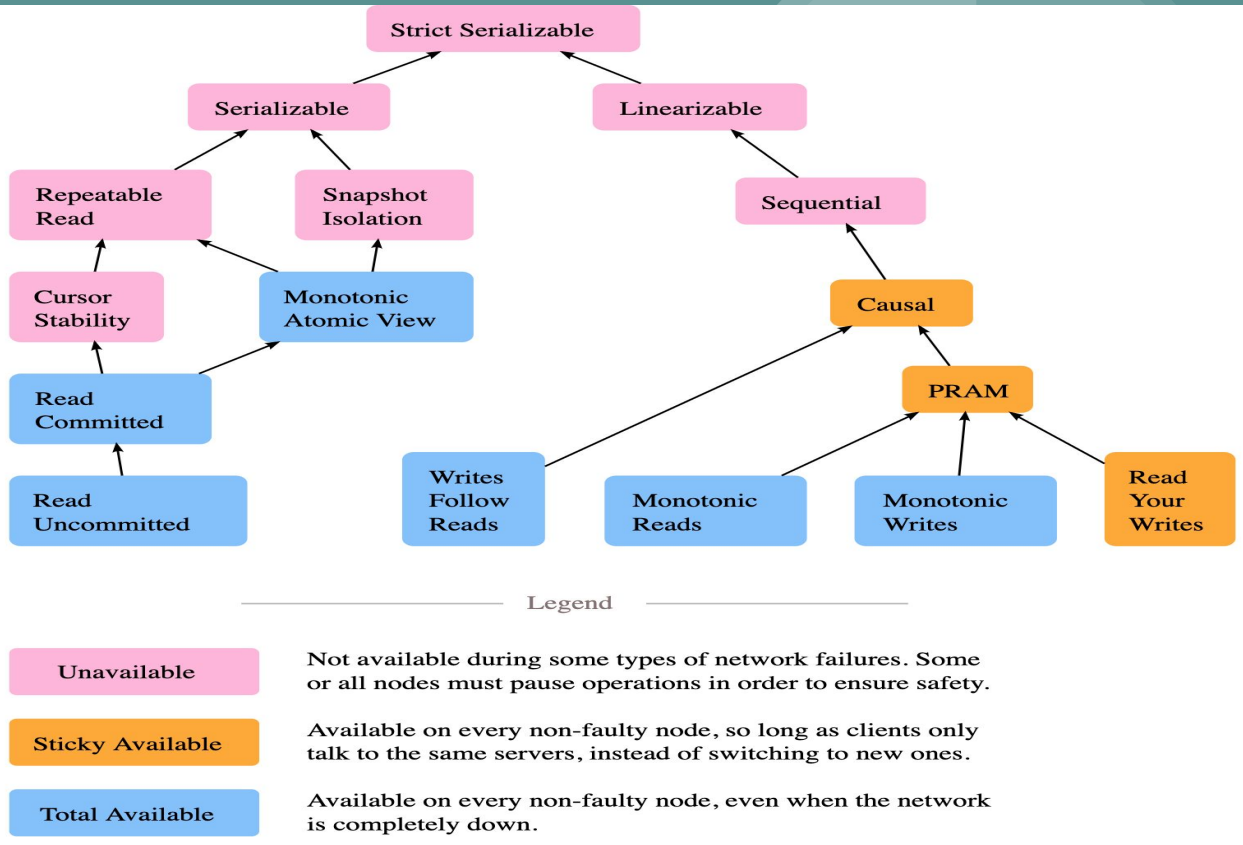
Partial Order



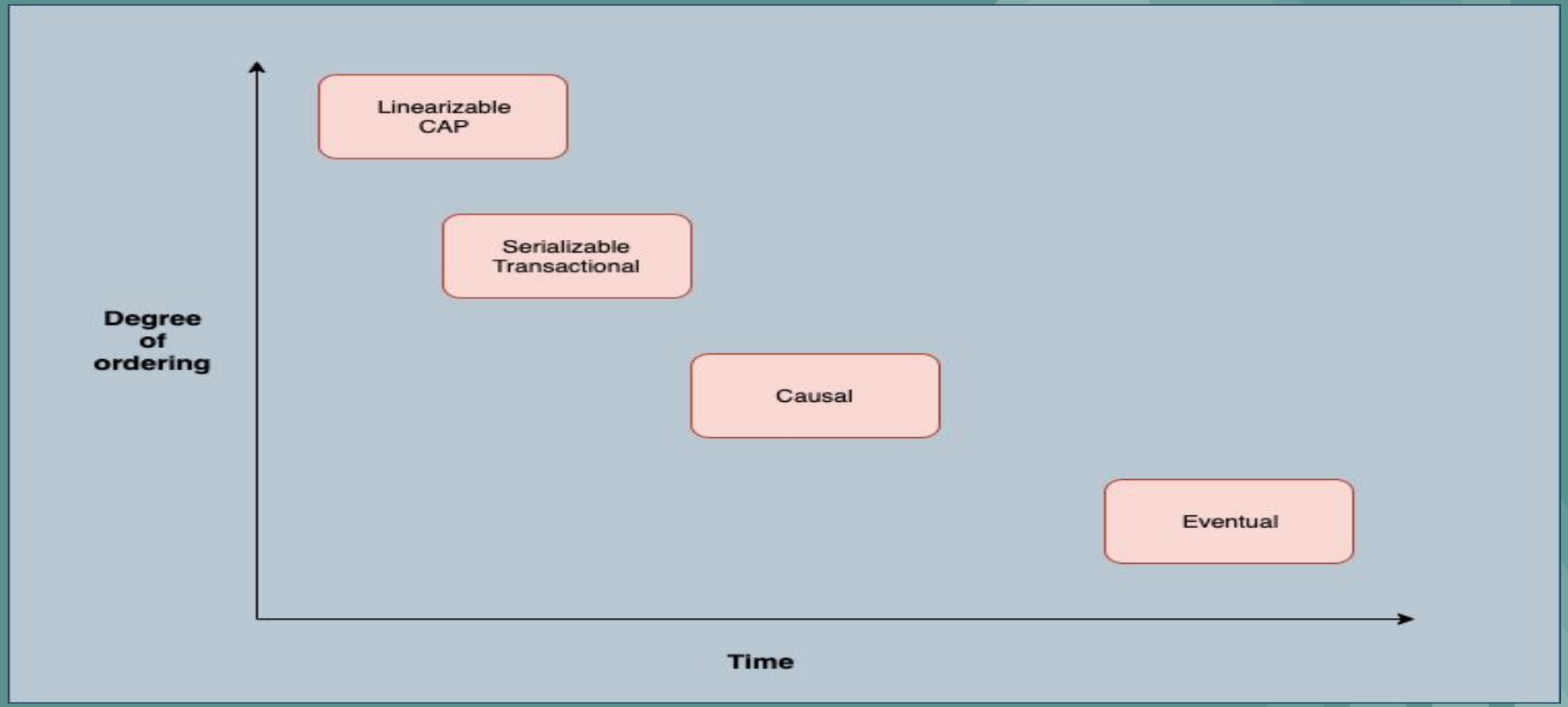
Consistency Models



Consistency Models



Consistency and Ordering

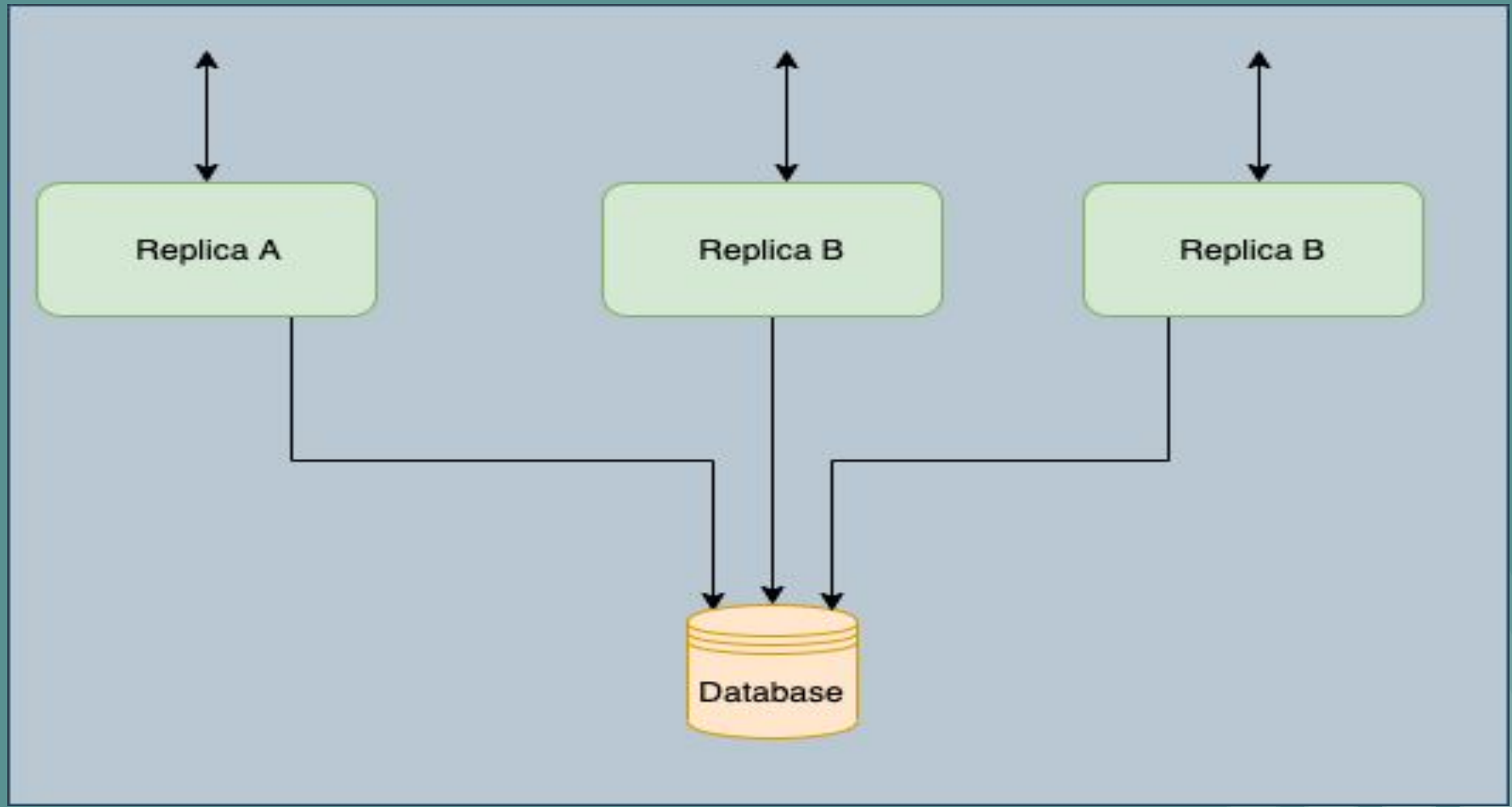


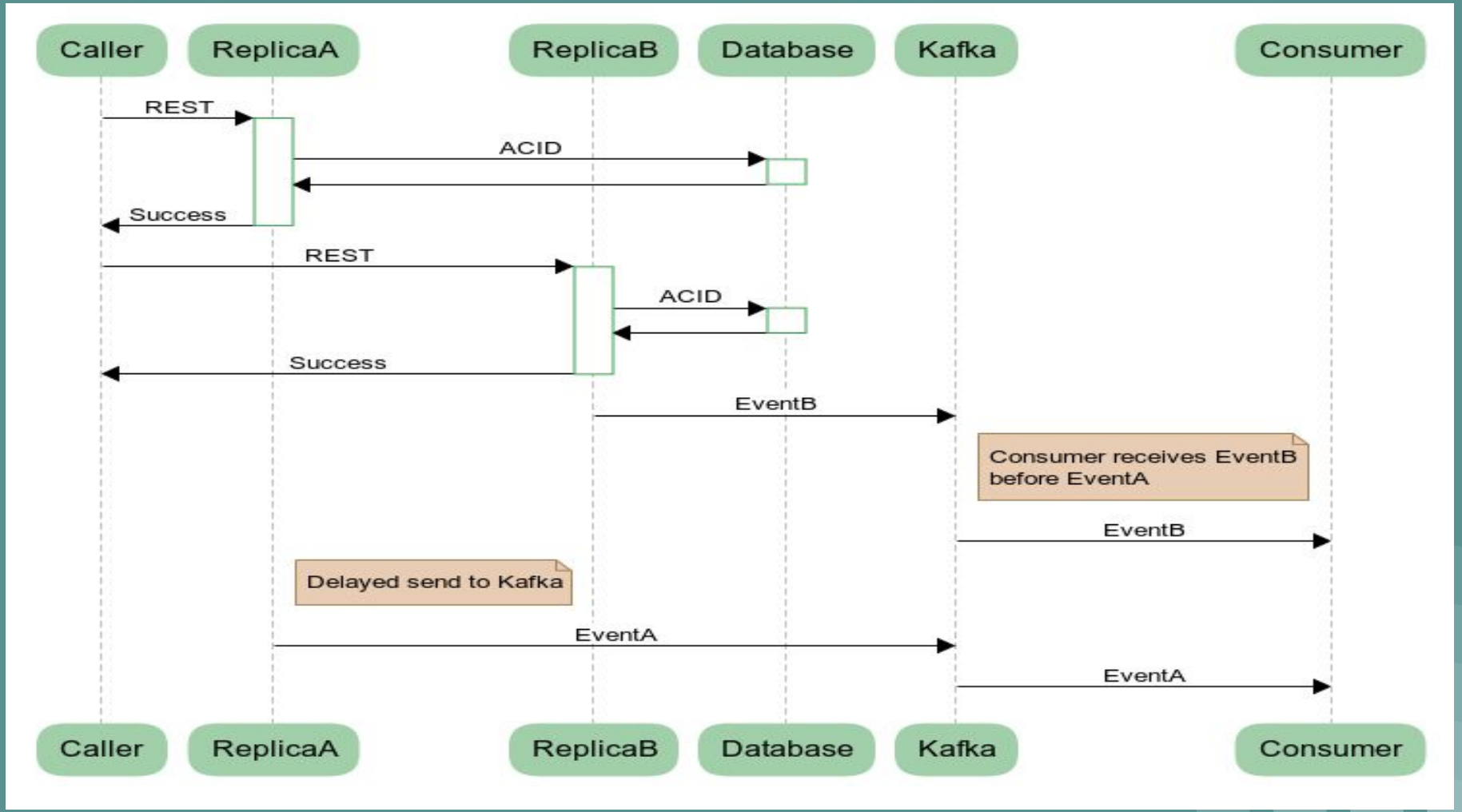
Consistency and Ordering

Consistency has a strong Equivalence to Ordering

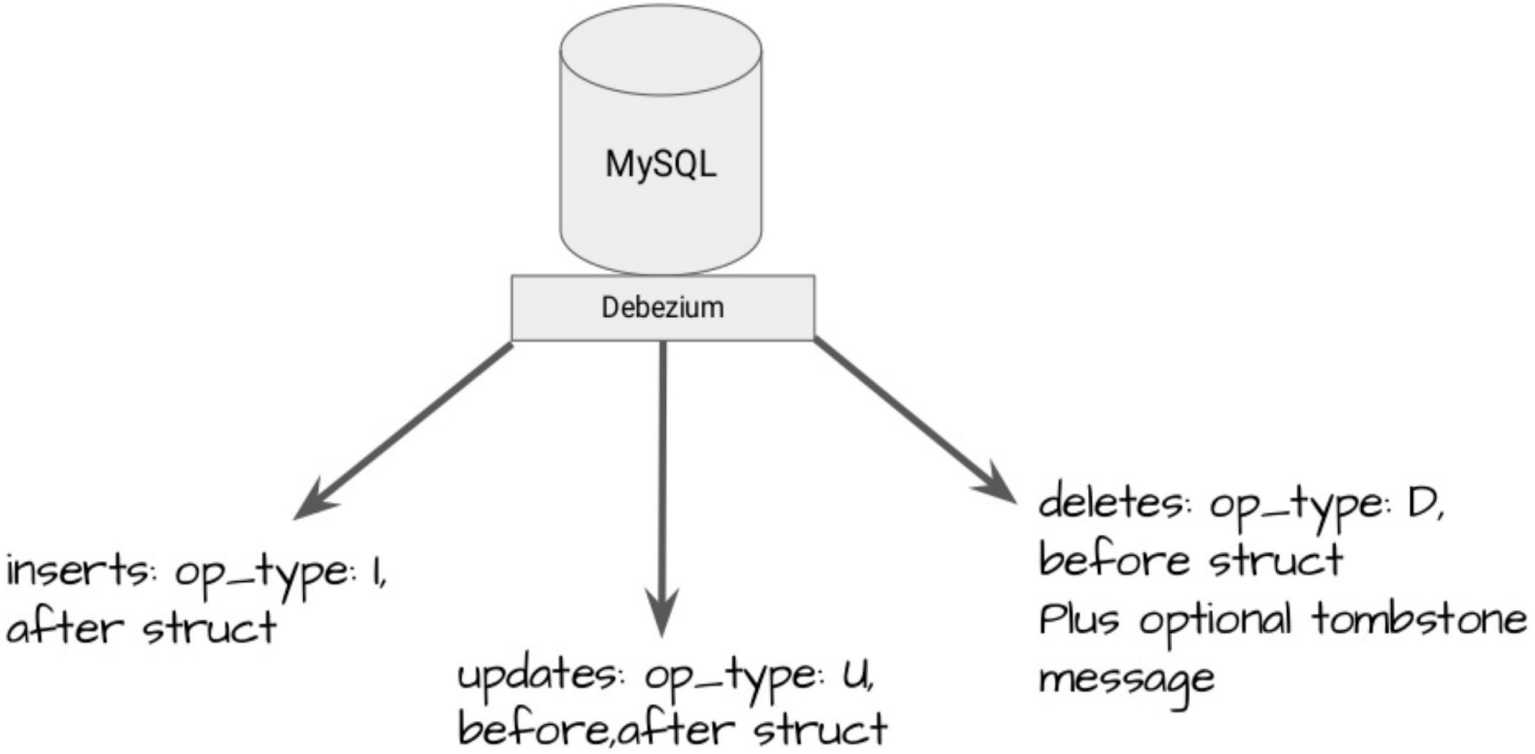
Academic Research

- Total Order Broadcast(Consensus) Algorithms
- CRDT's/CALM
- Workflow Completion
- Conflict Resolution (Distributed Graph Traversals/Causal and Vector clocks)


Replica A
Replica B
Replica B
Database


Caller
ReplicaA
ReplicaB
Database
Kafka
Consumer
REST
ACID
Success
REST
ACID
Success
EventB
Consumer receives EventB before EventA
EventB
Delayed send to Kafka
EventA
EventA
Caller
ReplicaA
ReplicaB
Database
Kafka
Consumer

CDC Overview

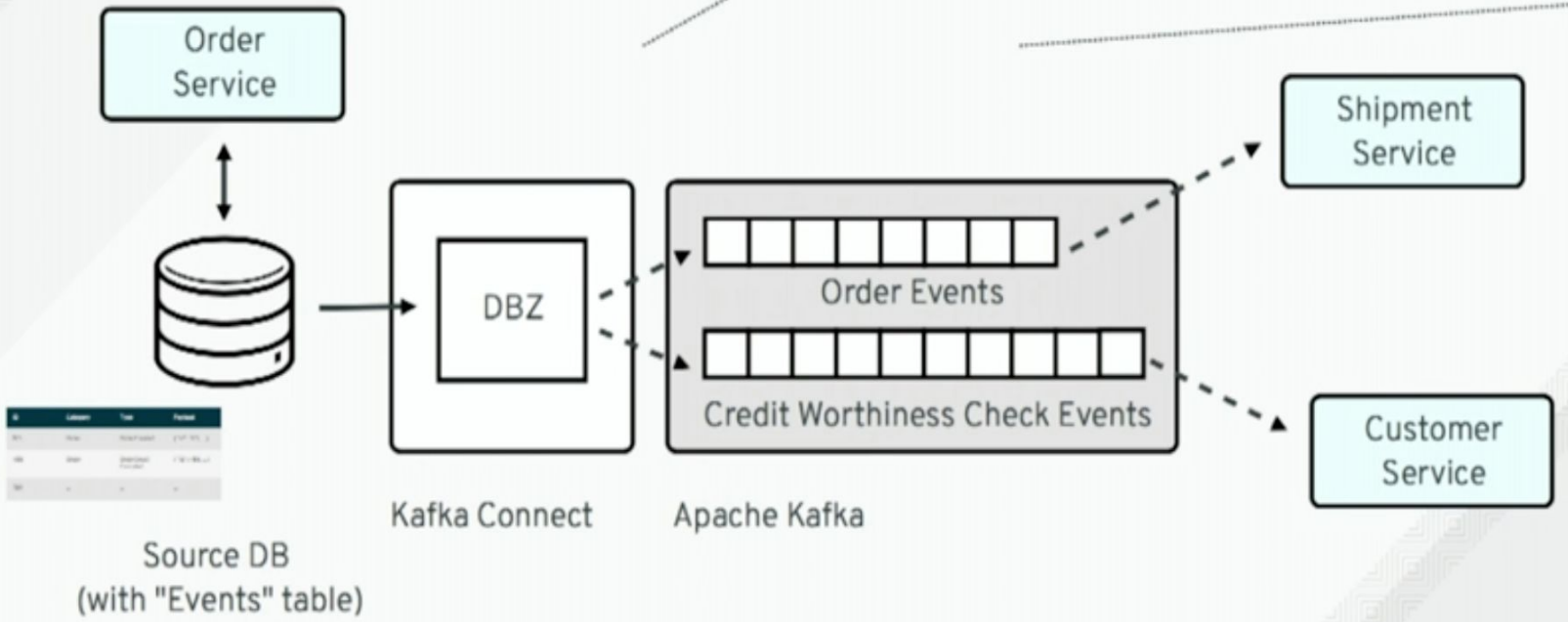


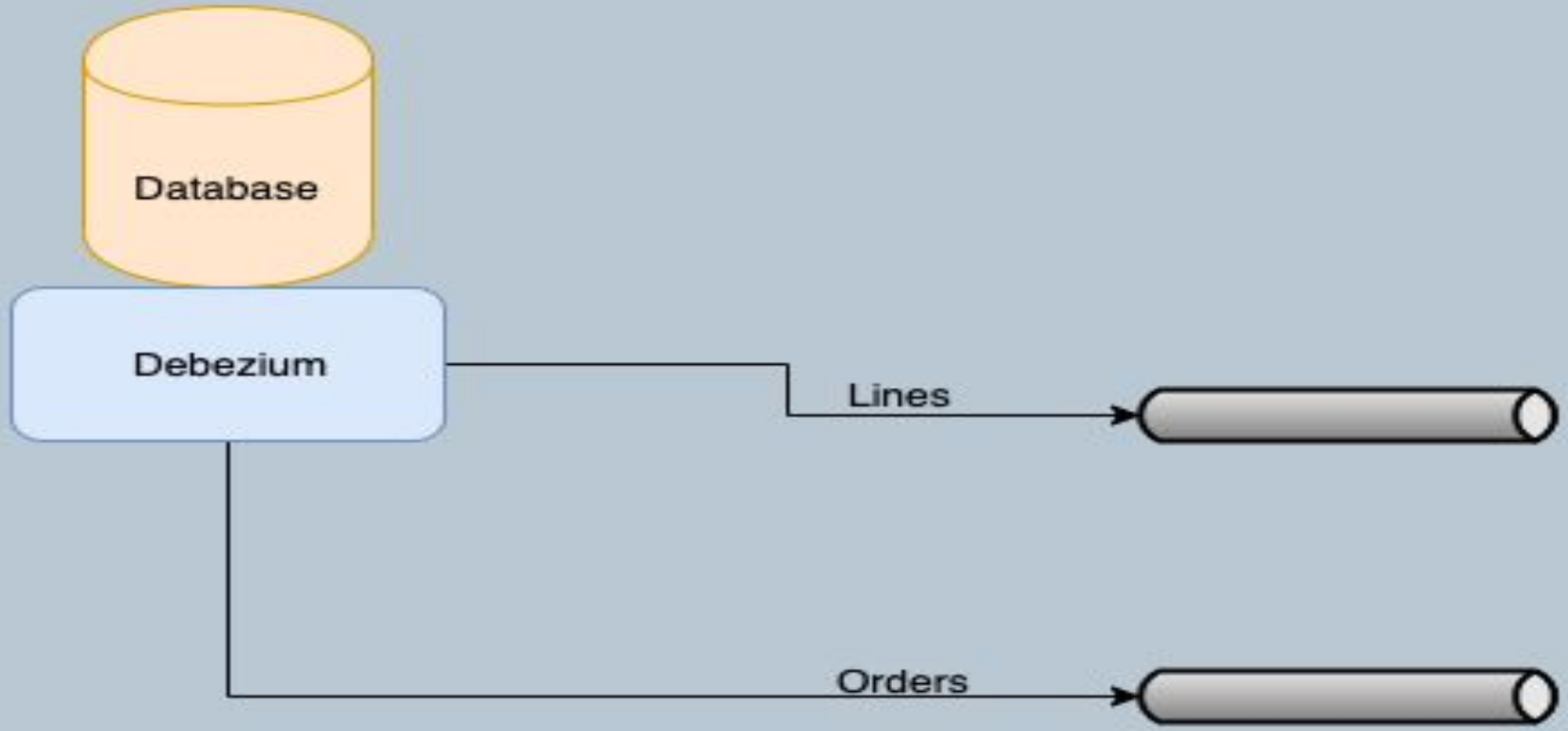
Pattern: Outbox

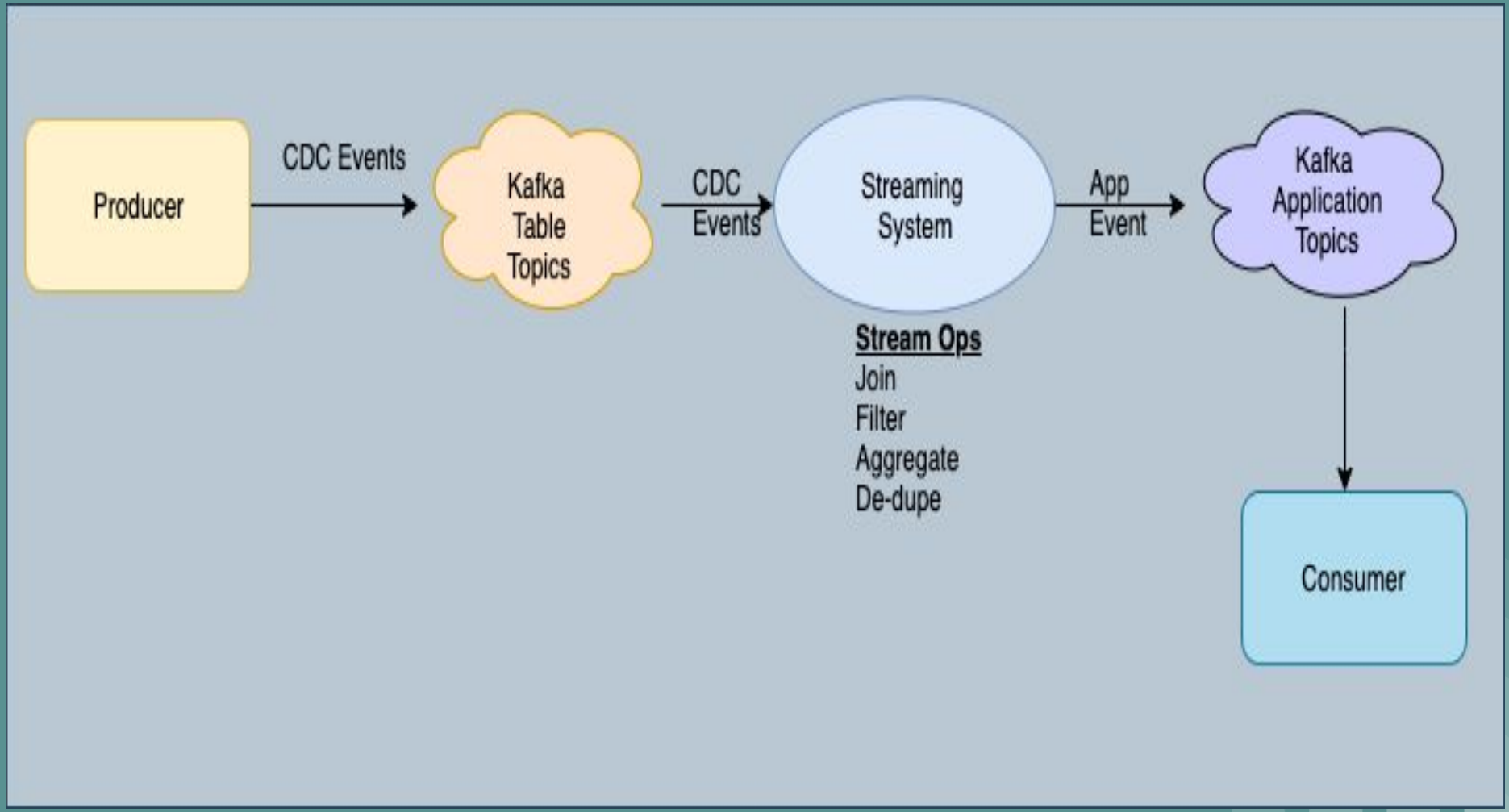
Separate Events Table

"Outbox" table

Id	AggregateType	AggregateId	Type	Payload
ec6e	Order	123	OrderCreated	{ "id" : 123, ... }
8af8	Order	456	OrderDetailCanceled	{ "id" : 456, ... }
890b	Customer	789	InvoiceCreated	{ "id" : 789, ... }

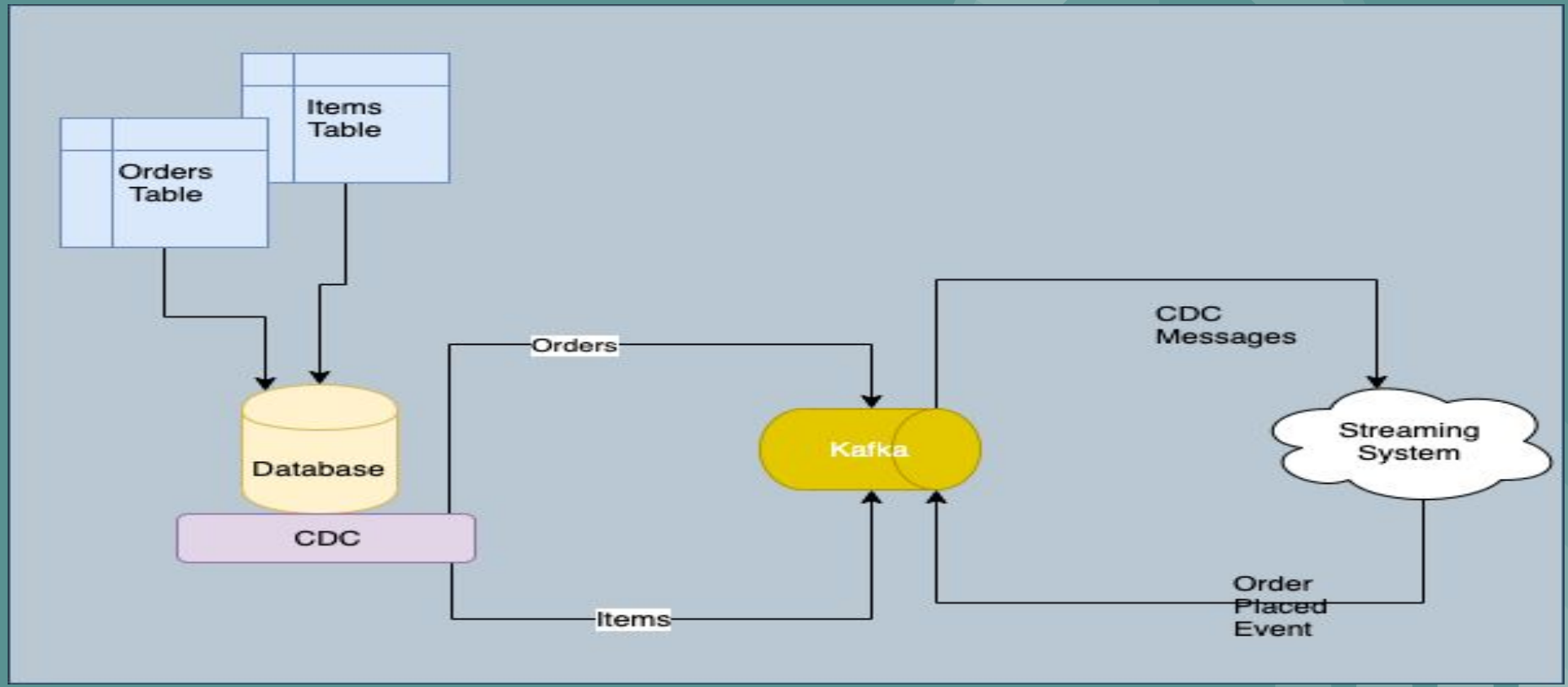



Database
Debezium
Lines
Orders


Producer
CDC Events
Kafka
Table
Topics
CDC
Events
Streaming
System
App
Event
Kafka
Application
Topics
Stream Ops
Join
Filter
Aggregate
De-dupe
Consumer

Event Completeness

- Consider an Order with id:101 and 3 items.



Event Completeness

groupBy + aggregate + join + filter

KTable<OrderId,ArrayList<Items>> preItemsTable = itemsStream<ItemId,Items>

1. .groupBy(OrderId) → we want to collect things per order.
2. .aggregate(ArrayList::new, add(Items))
3. .join(totalNumberOfItems) → allow propagation of total items

KTable<OrderId,ArrayList<Items>> fullItemsTable = preItemsTable

1. .filter((k,v) -> v.size() == v.get(0).get("itemCount").asInt()) → This filter will block till until all the items for that order are in the array.

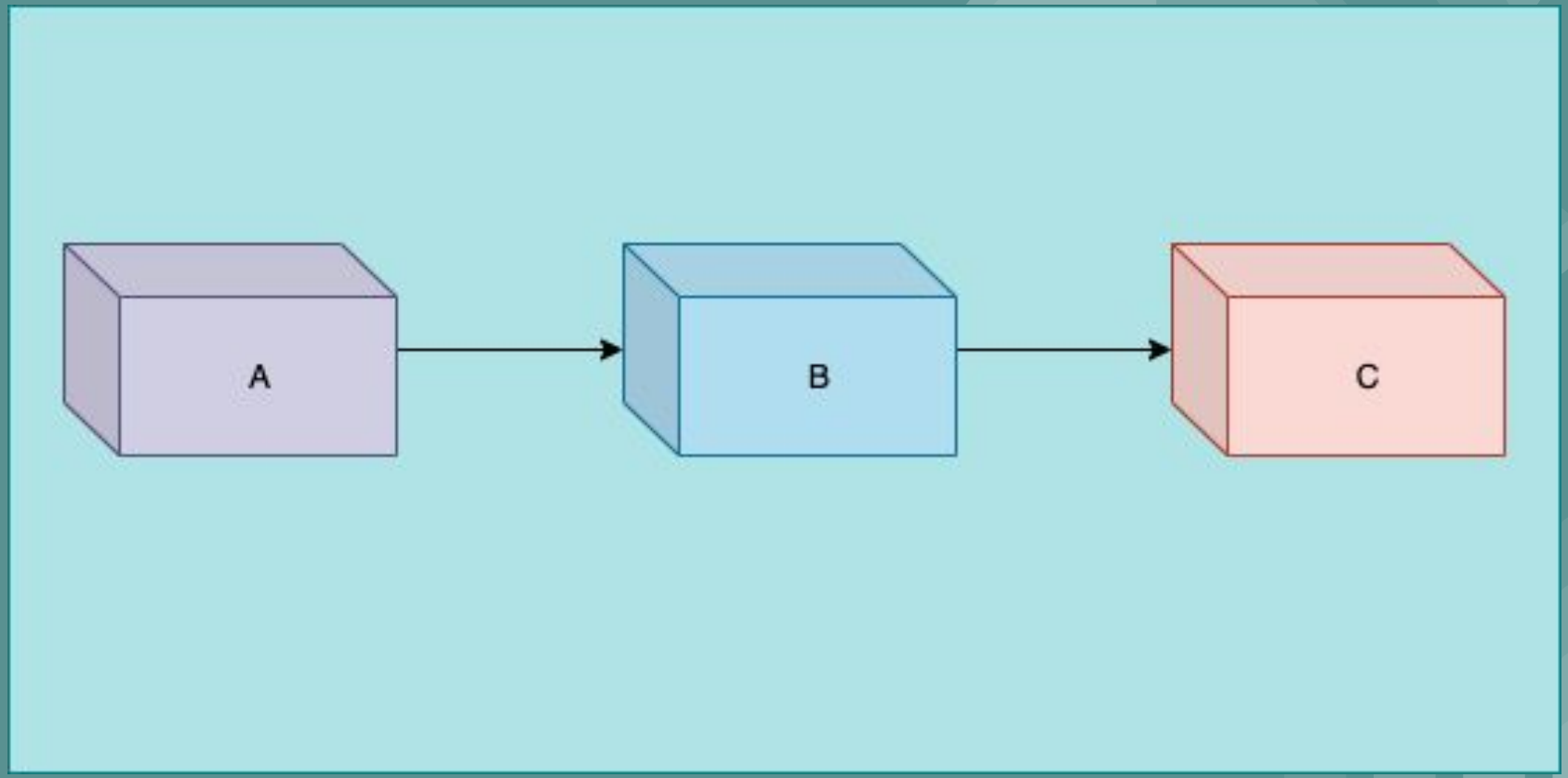
Event Completeness

Create the order placed Event

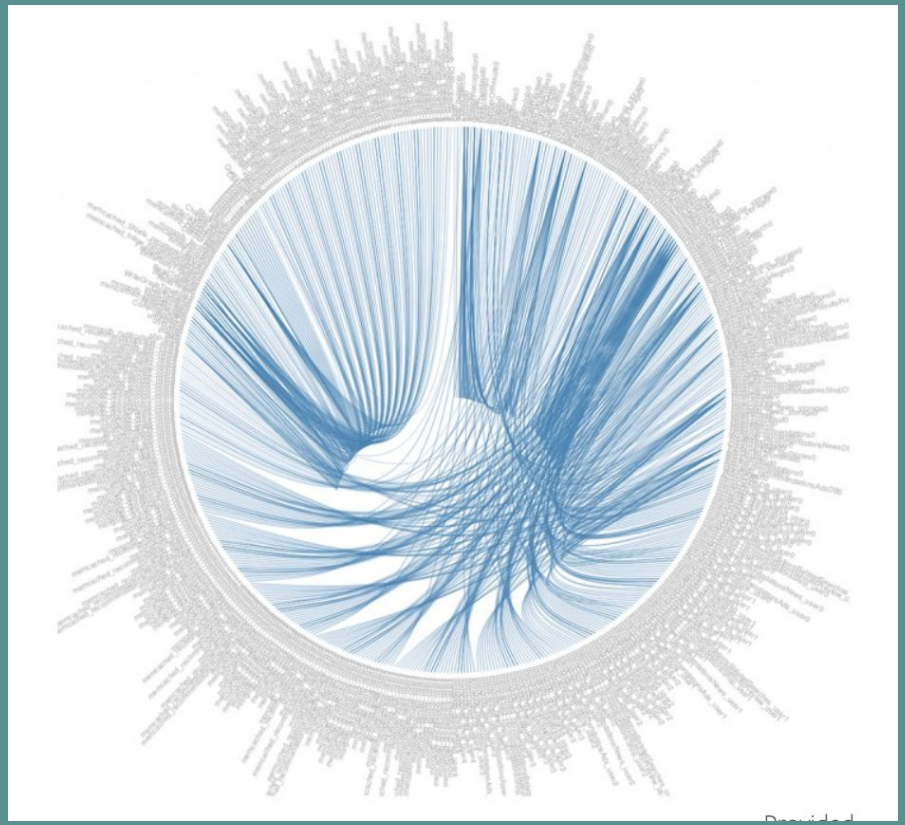
```
KTable fullOrderTable = orderTable
.join(fullItemsTable, (orderNode, itemNodes) -> {
        //construct and return order placed result
}
```

Complete example available at https://github.com/maneeshchaturvedi/events

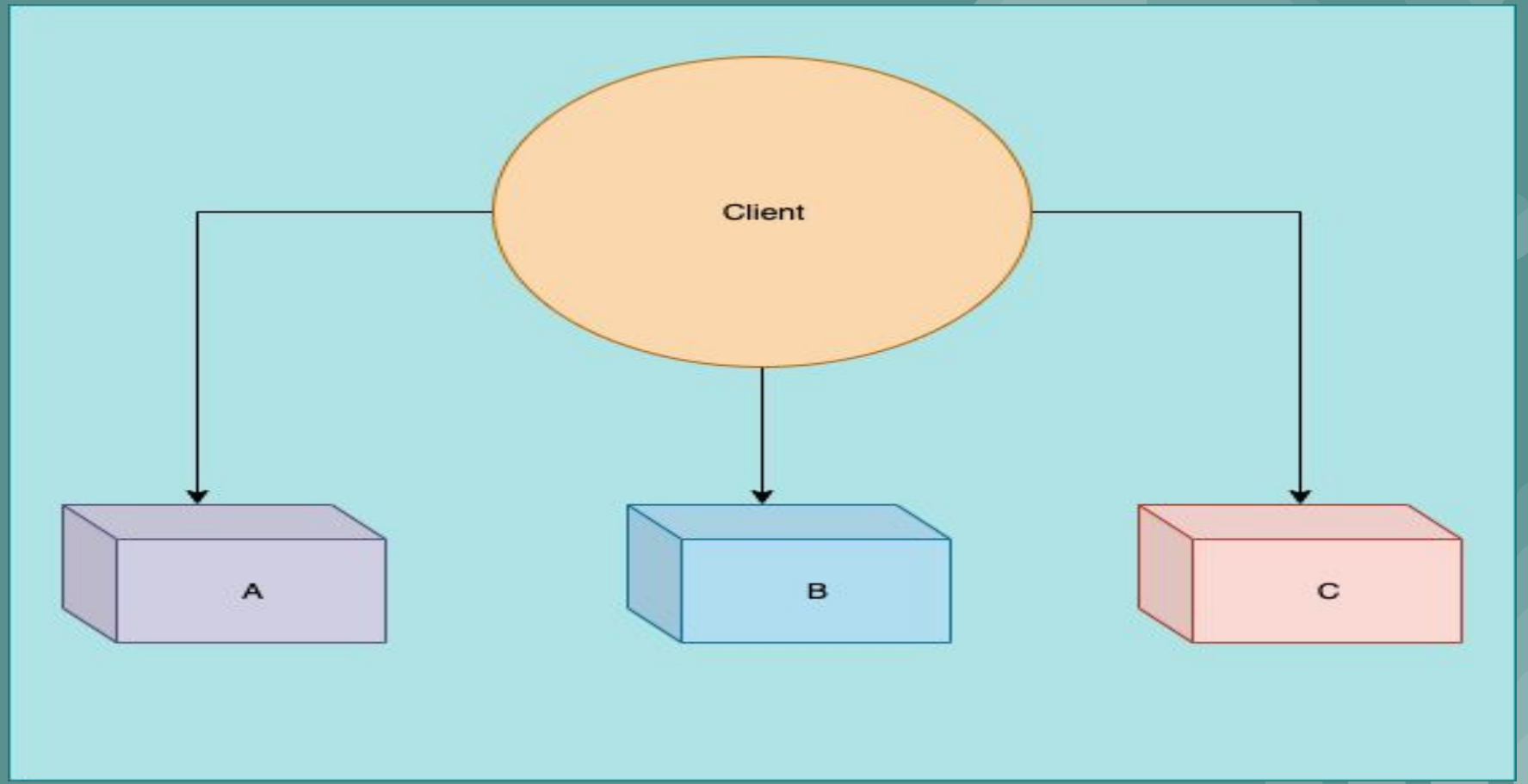
Functional Decomposition



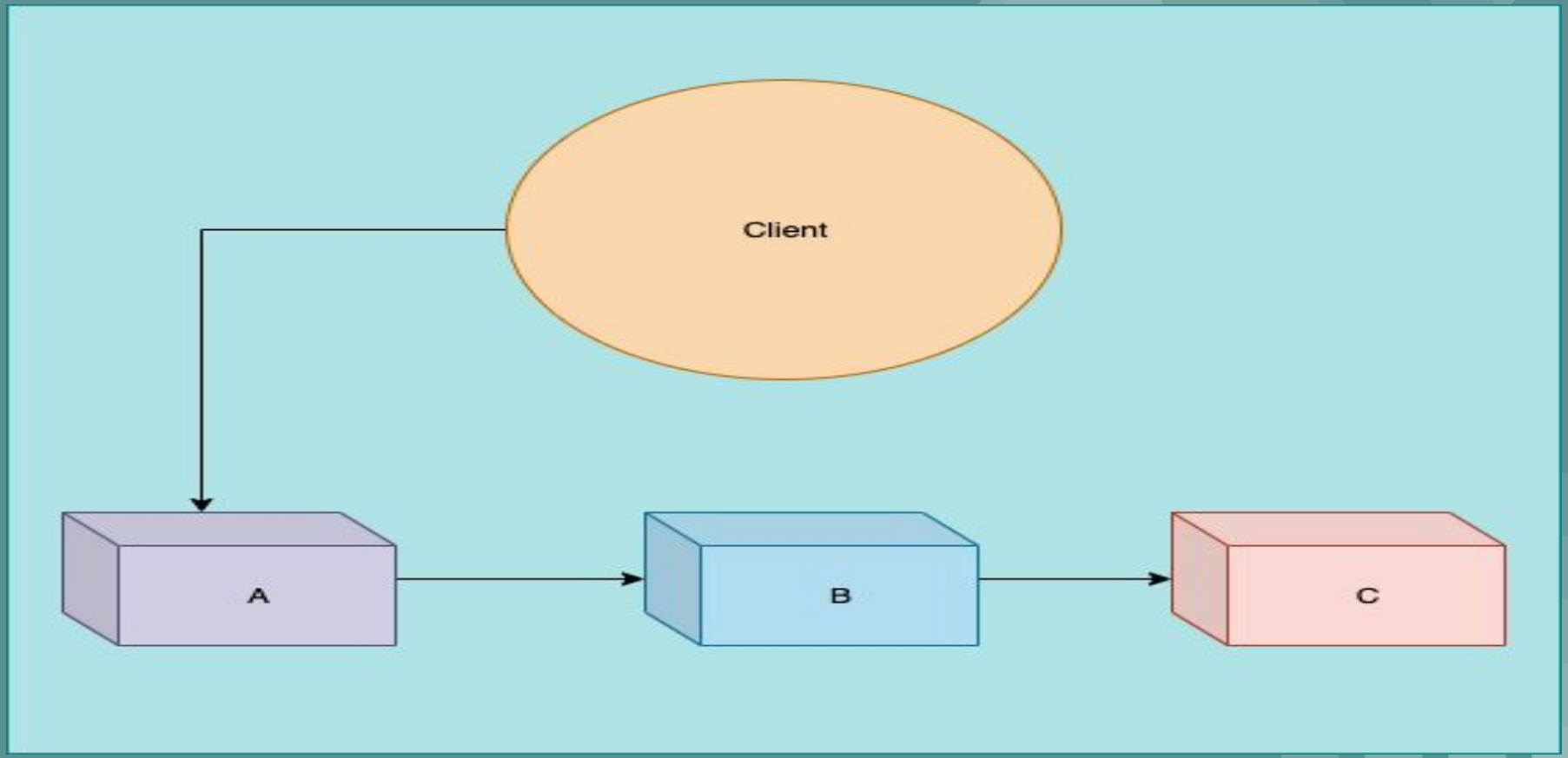
Functional Decomposition



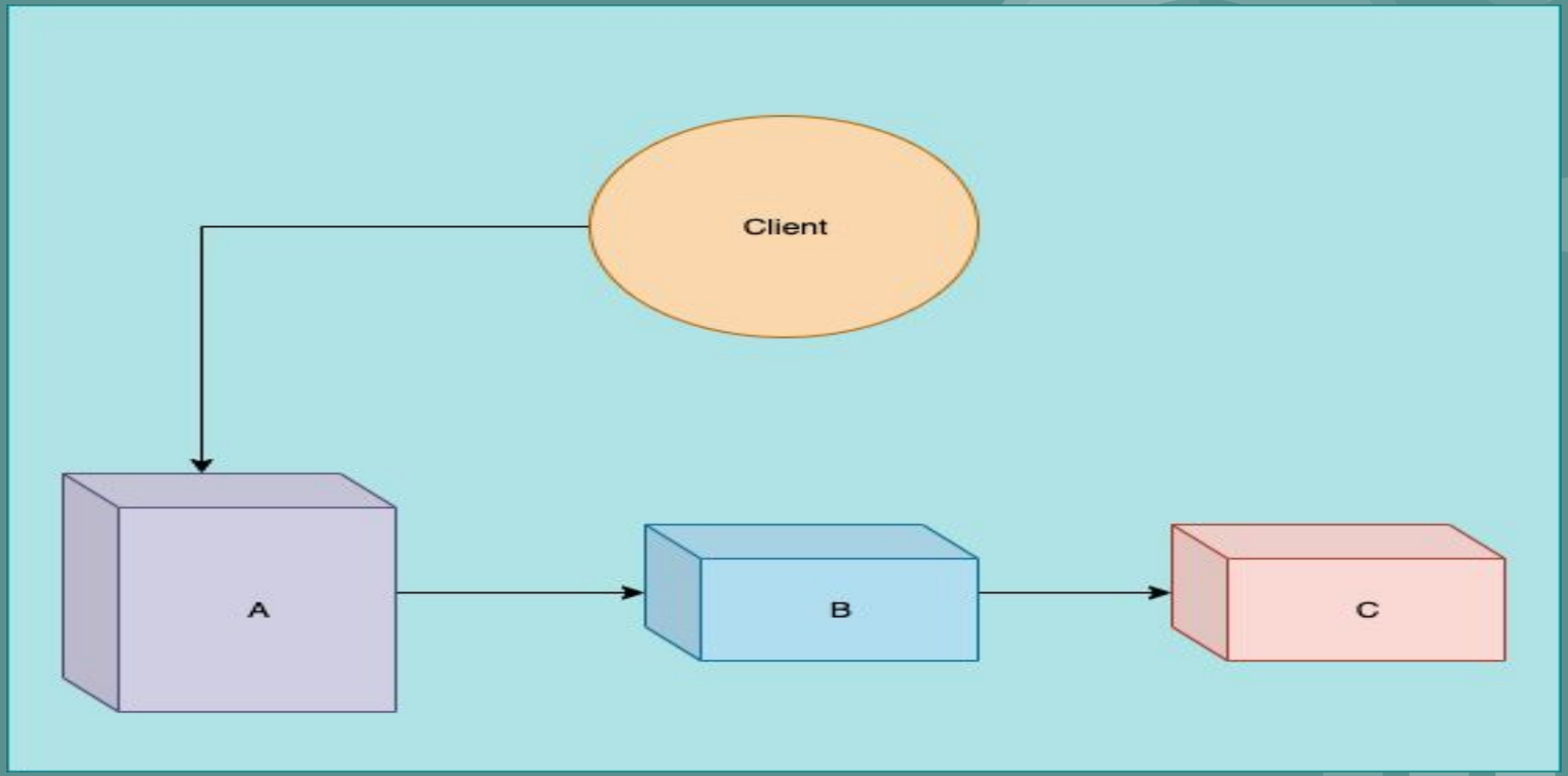
Functional Decomposition



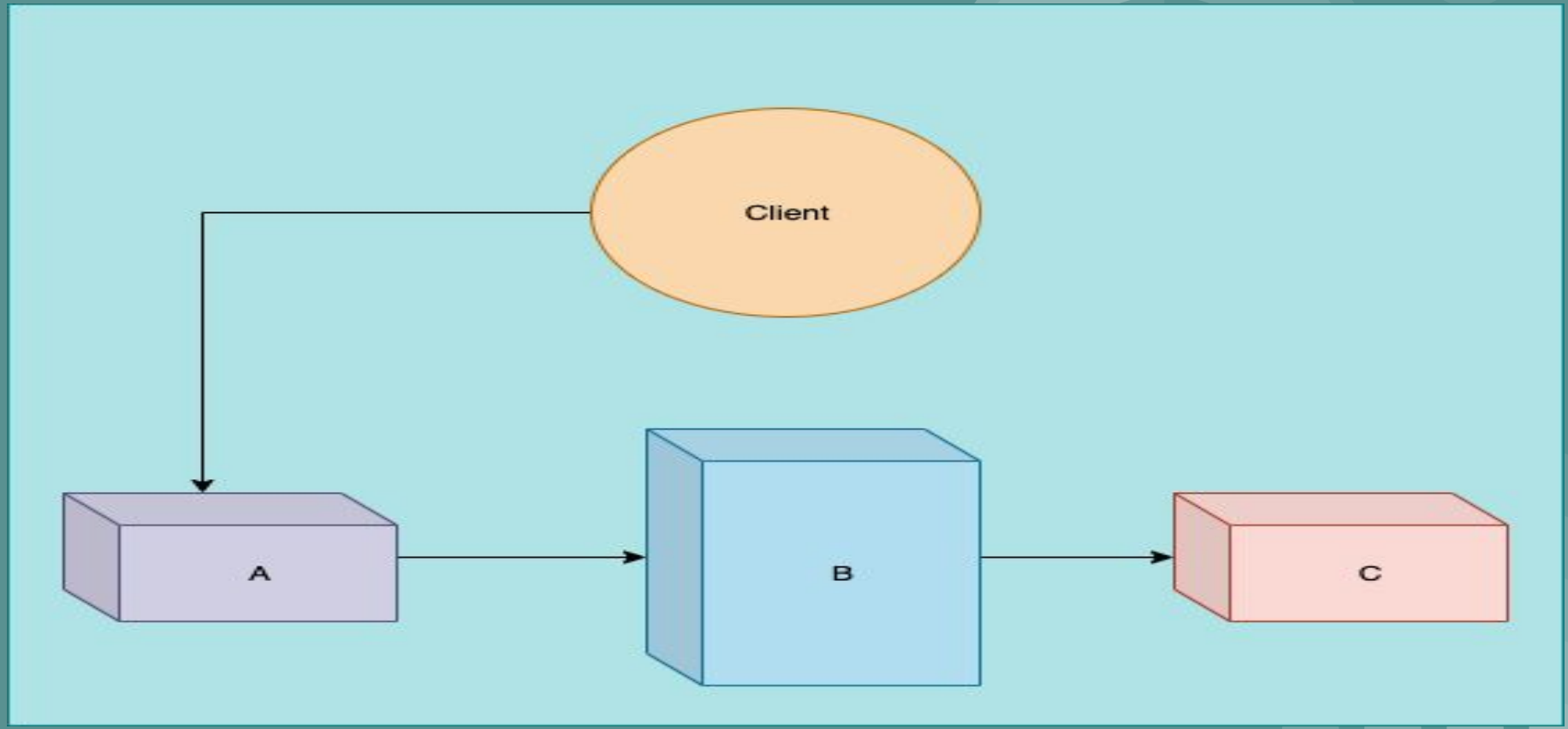
Functional Decomposition



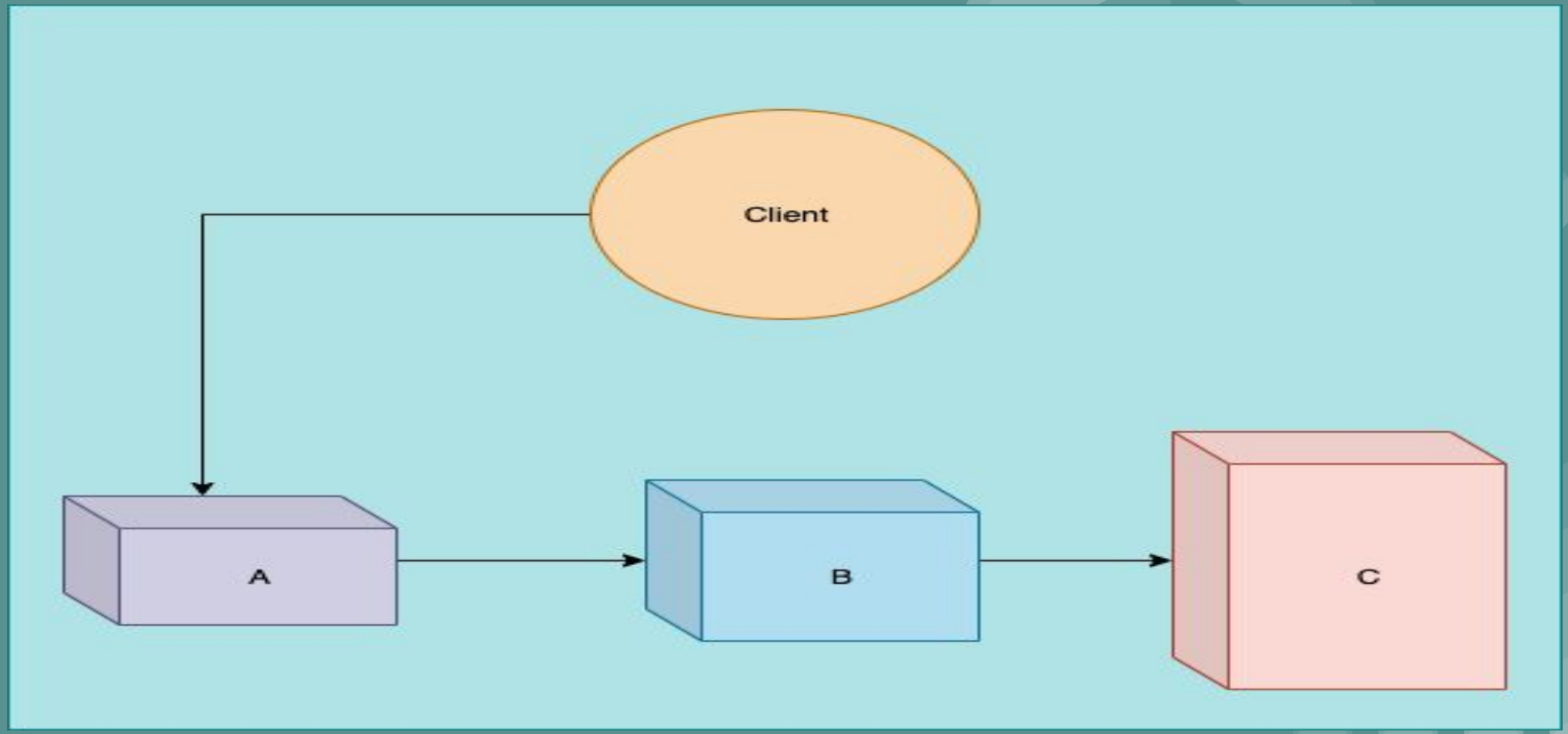
Functional Decomposition



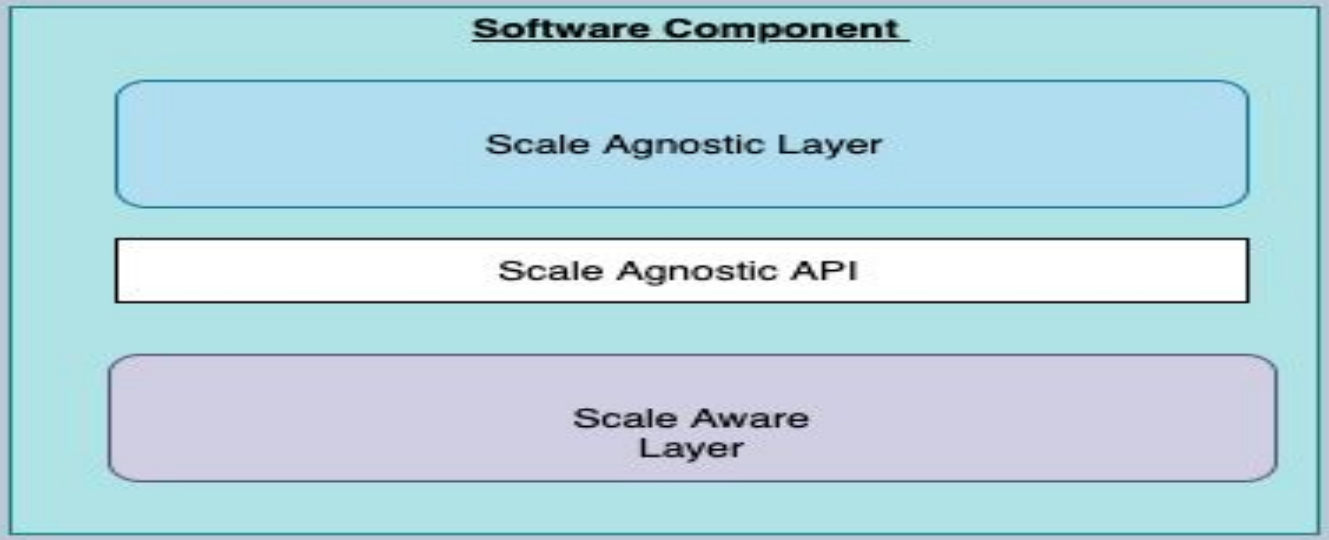
Functional Decomposition



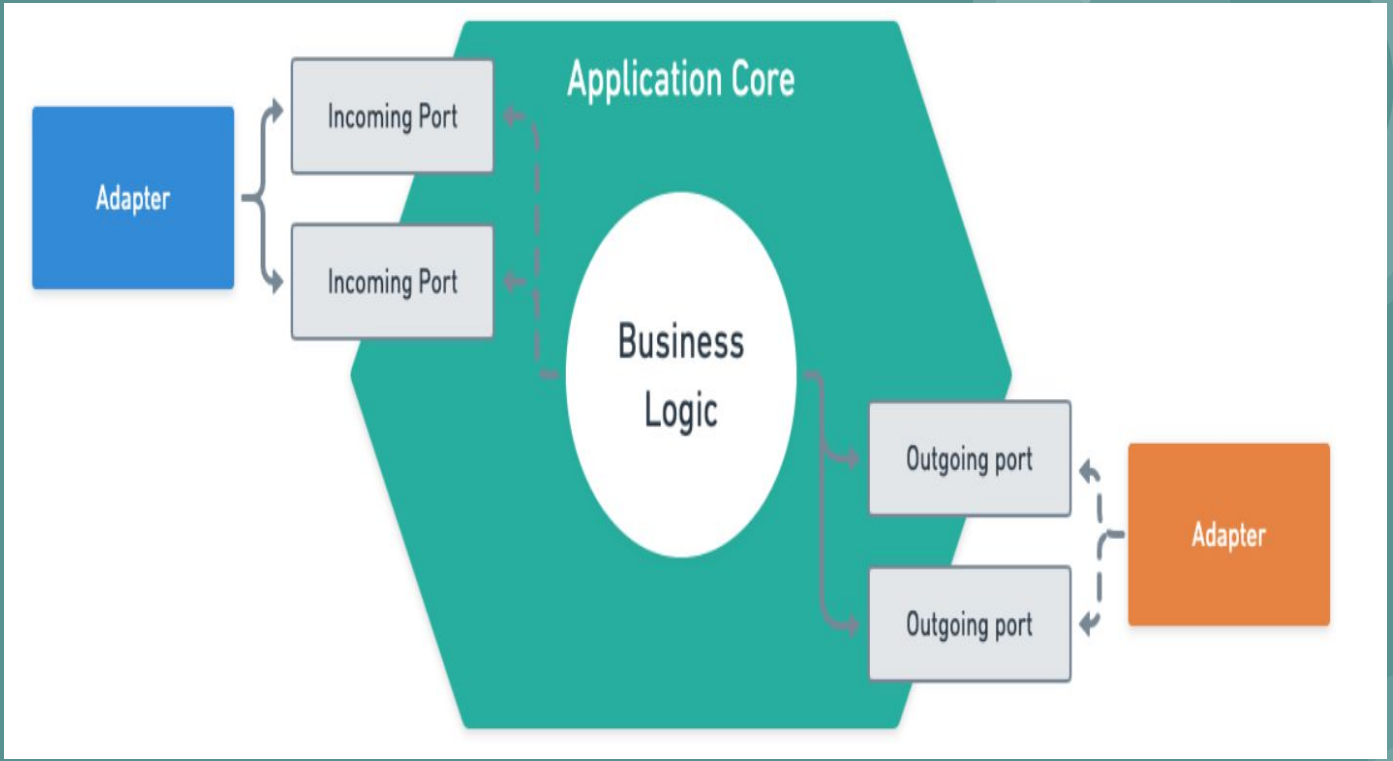
Functional Decomposition



Separation Of Concerns



Hexagonal Architecture



Quotes To Live By

When a design decision can reasonably go one of two ways, an architect needs to take a step back. Instead of trying to decide between options A and B, the question becomes "How do I design so that the choice between A and B is less significant?" The most interesting thing is not actually the choice between A and B, but the fact that there is a choice between A and B.
Kevlin Henney

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

Quotes To Live By

Programming is a design activity.
Jack W Reeves : http://user.it.uu.se/~carle/softcraft/notes/Reeve_SourceCodeIsTheDesign.pdf

You have a problem. You decide to solve it with configuration.
Now you have <%= $problems %> problems!
Dan North

People overvalue their knowledge and underestimate the probability of their being wrong.

If you have a procedure with ten parameters, you probably missed some.
Alan Perlis

Quotes To Live By

- Software development is recursive. You build large systems that are software, these are built from smaller things which are software which in turn are built from even smaller things which are again software. Even one badly written construct can affect the architecture as a whole.
- For the beginner architect, there are many options
  For the master architect, there are but a few.

Gracias
Preguntas ??

References

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing
https://arxiv.org/pdf/1512.00168.pdf
https://lamport.azurewebsites.net/pubs/time-clocks.pdf
https://dsf.berkeley.edu/jmh/calm-cidr-short.pdf
http://bloom-lang.net/calm/
http://www.vldb.org/pvldb/vol7/p181-bailis.pdf
https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type
https://jepsen.io/consistency
https://en.wikipedia.org/wiki/Linearizability
https://en.wikipedia.org/wiki/Serializability
https://en.wikipedia.org/wiki/Eventual_consistency