

# Just-right consistency: Antidote

Guest lecture

Peter Van Roy  
Christopher Meiklejohn  
**Annette Bieniusa**

# Outline

**Part I:** Consistency in geo-replicated data stores

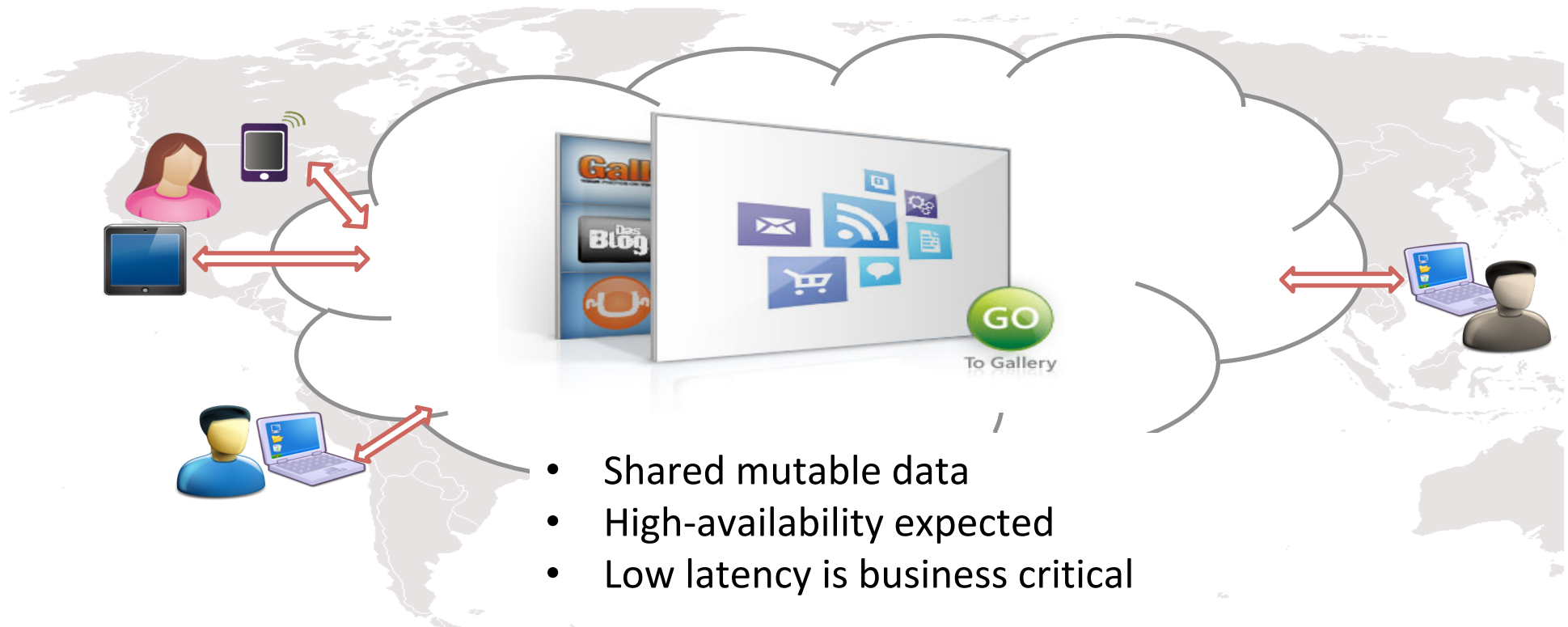
**Part II:** Consistency and invariant preservation

**Part III:** Antidote

# Part I

## Consistency in geo-replicated data stores

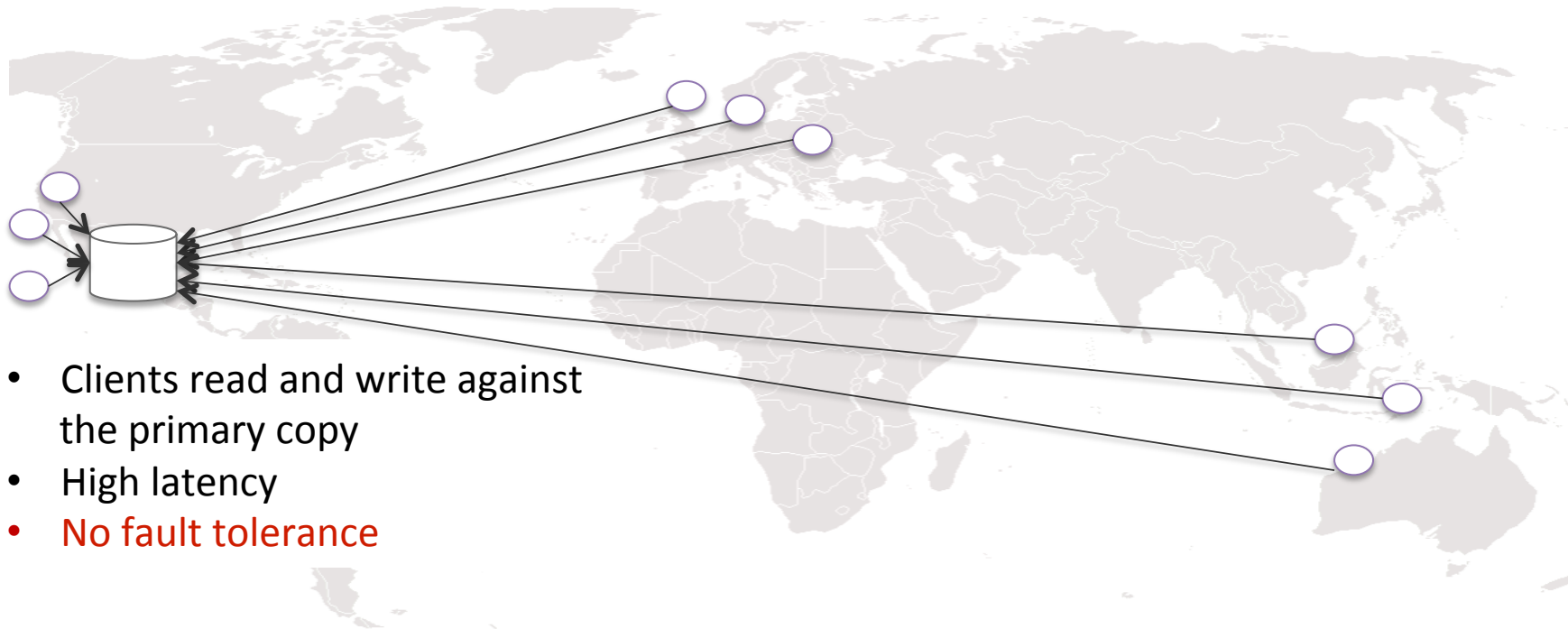
# Interactive distributed applications



# Cloud Databases

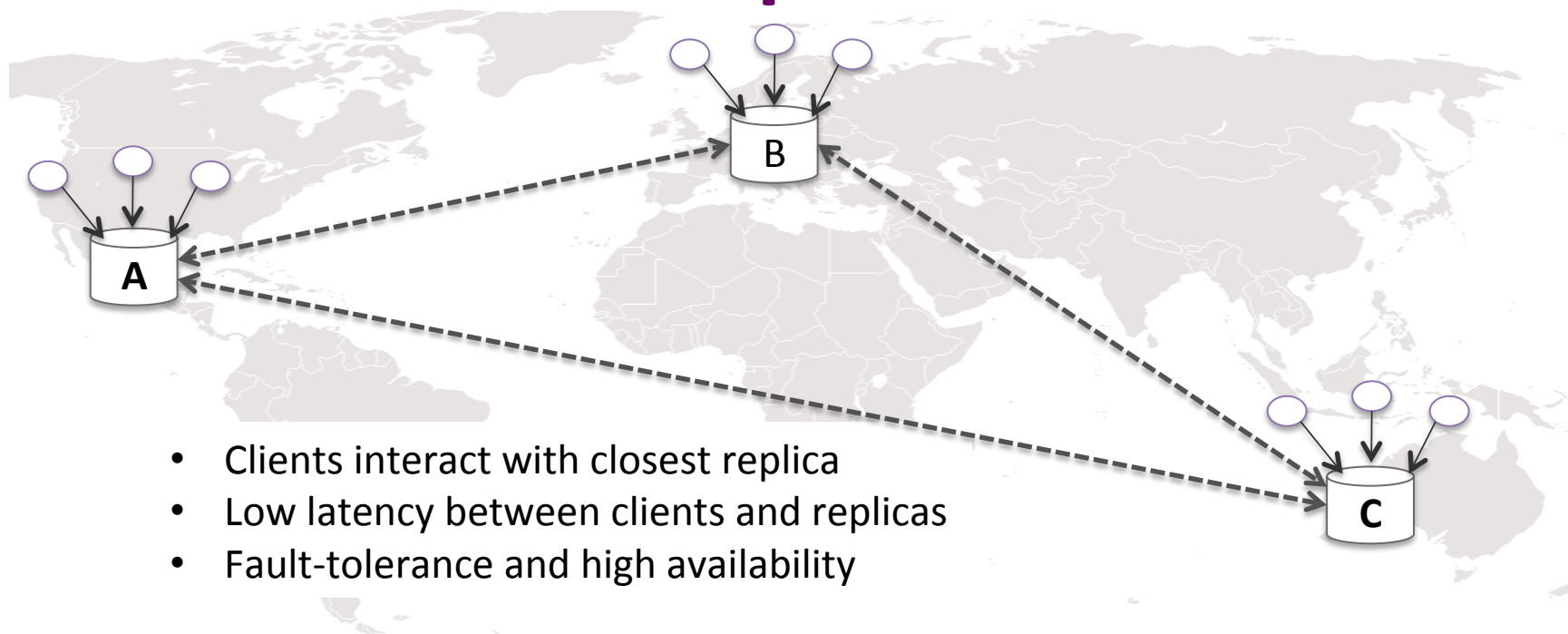


# Cloud Databases: Centralized deployment

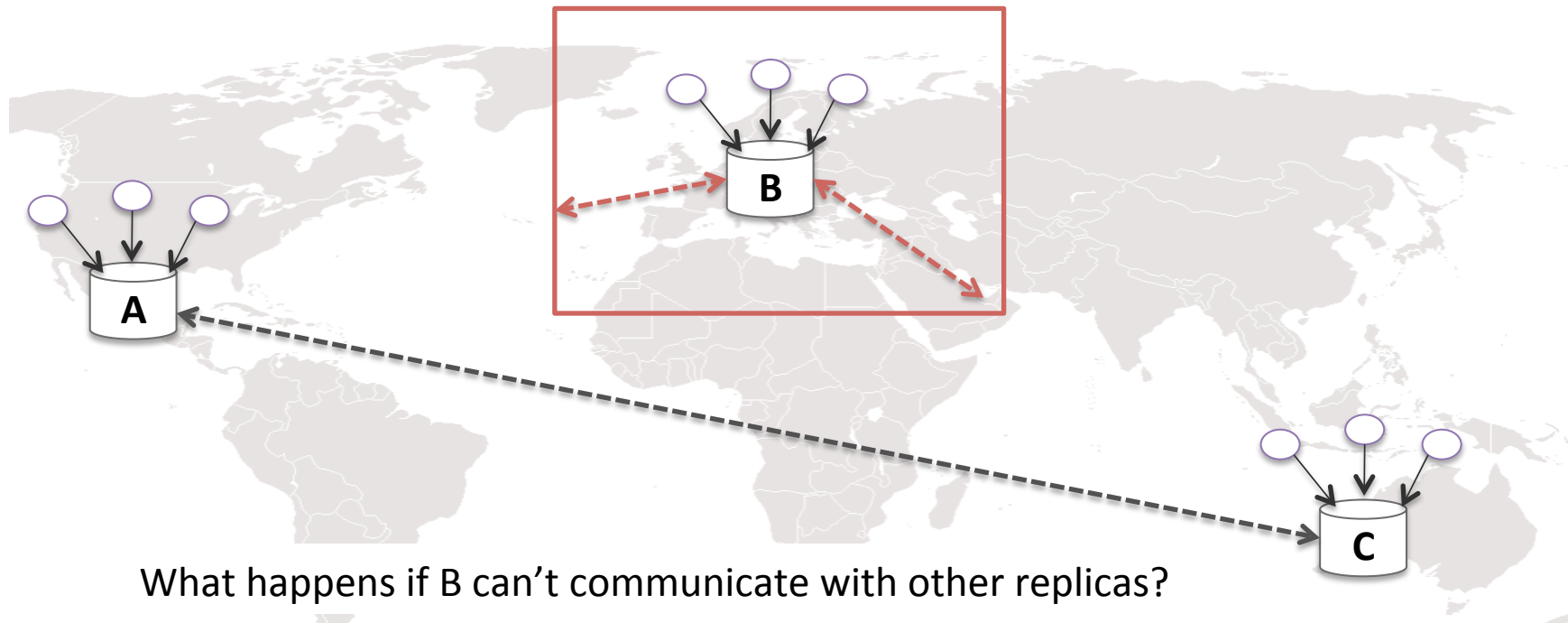


- Clients read and write against the primary copy
- High latency
- **No fault tolerance**

# Cloud Databases: Geo-replication

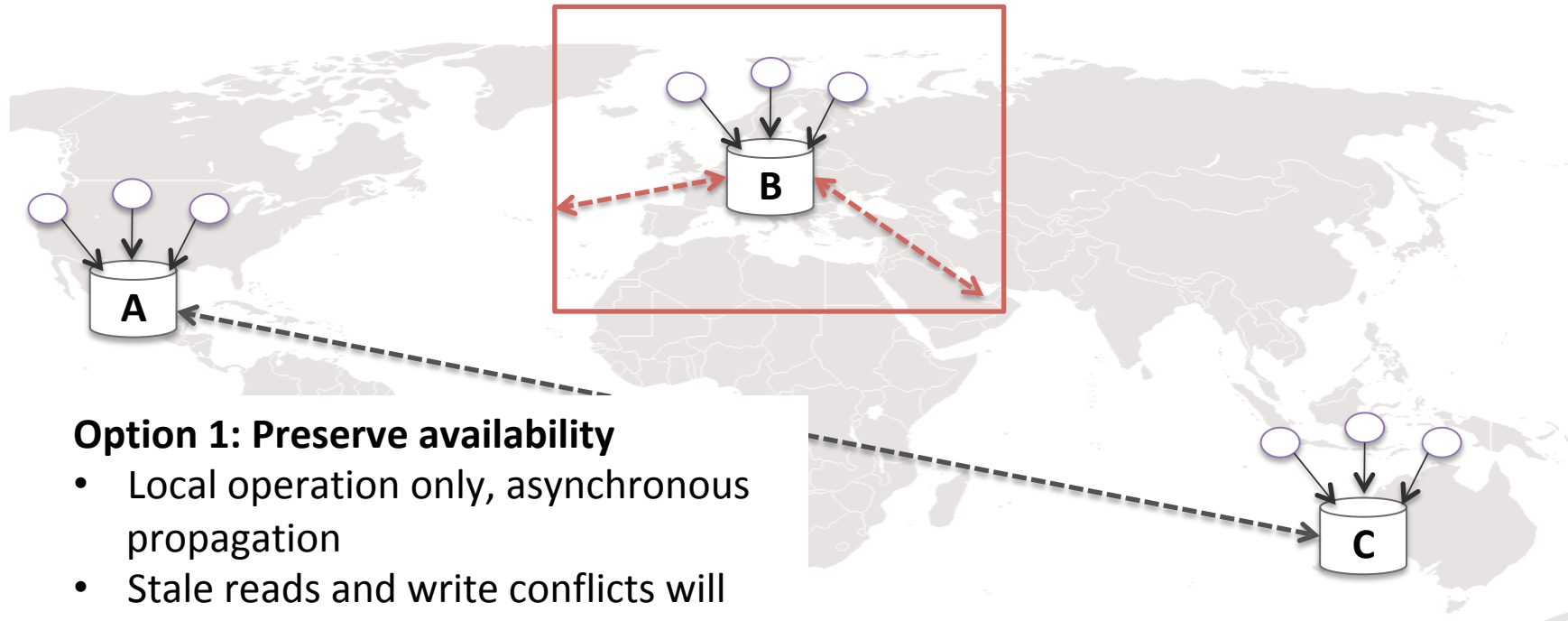


# Cloud Databases





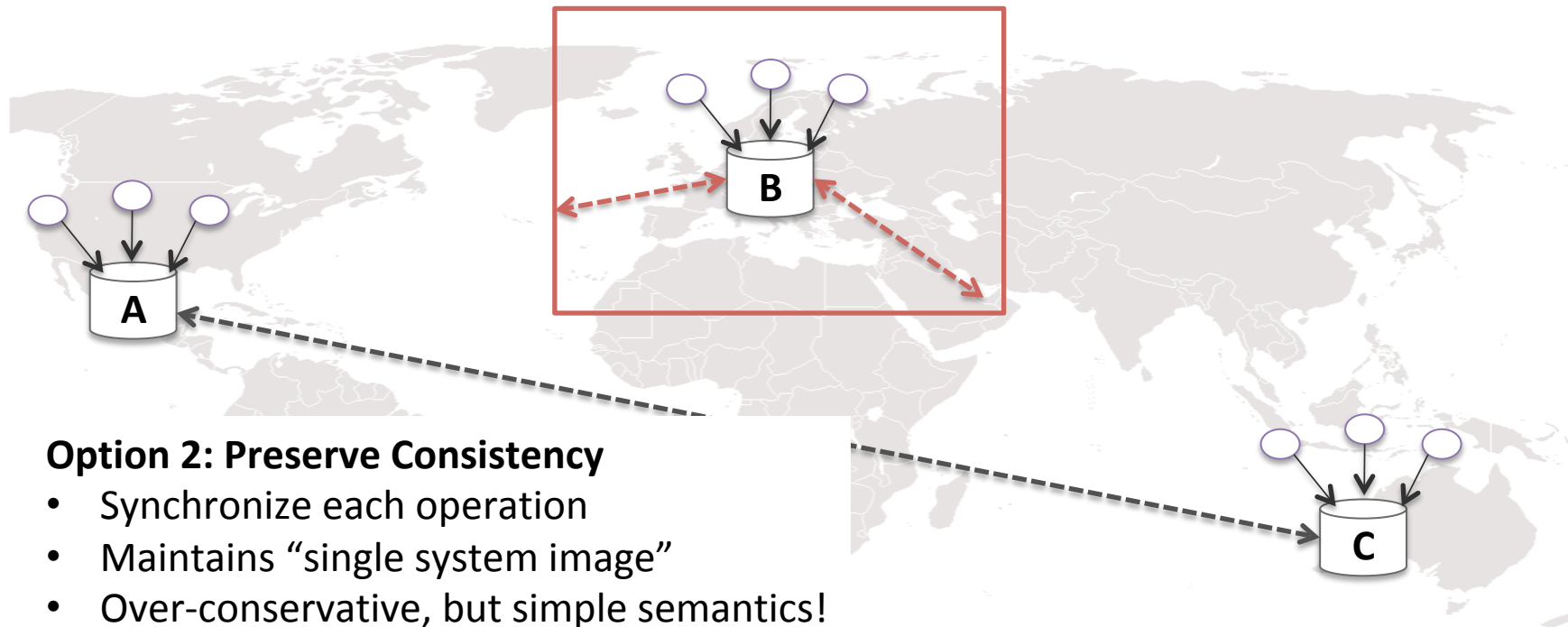
# Cloud Databases



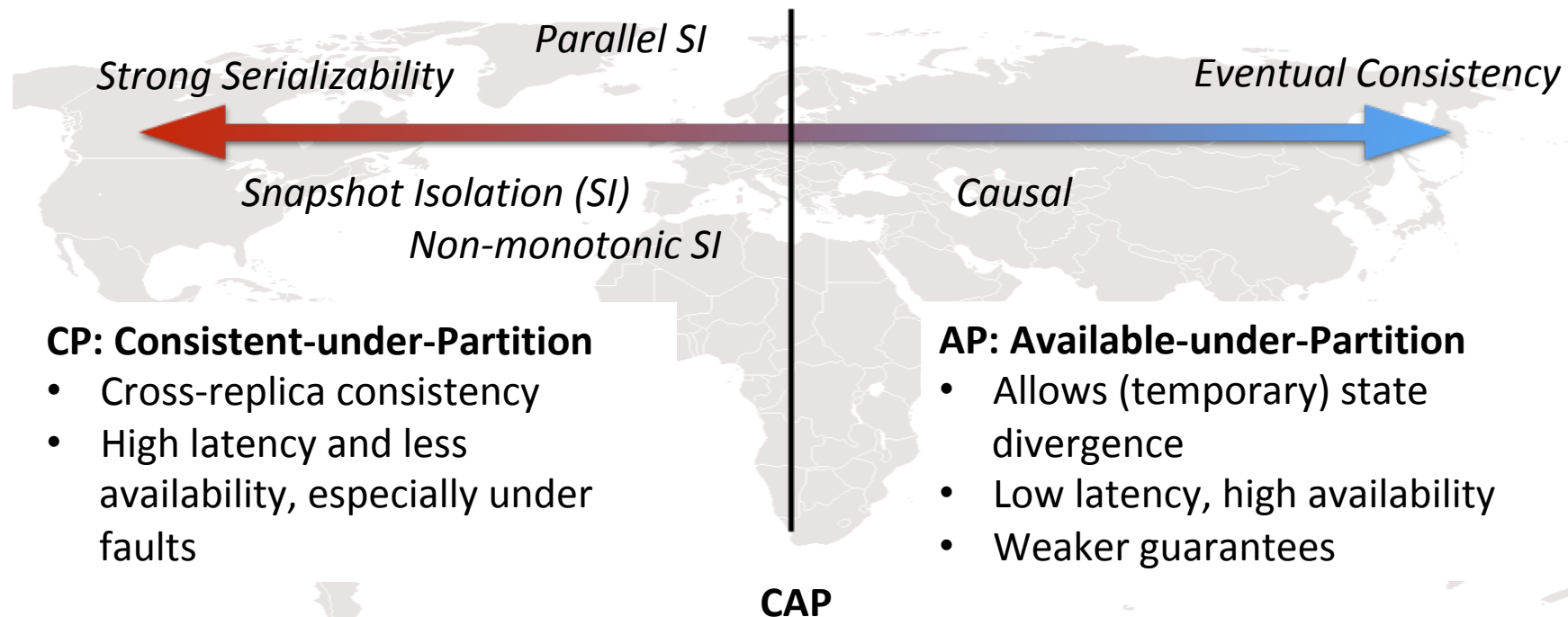
## Option 1: Preserve availability

- Local operation only, asynchronous propagation
- Stale reads and write conflicts will occur without synchronization

# Cloud Databases



# Consistency-Availability trade-off



# Cloud storage: AP systems

- To achieve low latency, high availability and throughput, systems have to forego strong consistency



*cassandra*



- Complex semantics
- Low-level programming interface
  - Key-value map
- No transactional support
- No relational mappings

# Cloud storage: CP Systems

- Cloud provides rely on expensive infrastructure to provide more guarantees



Google Cloud Platform



Azure Cosmos DB

- Strong consistency
- Support for transactions and SQL queries
- Coordination across sites
  - ... still high latency

# Alternative: AntidoteDB

- **AP** data store
- Provides strongest form of consistency that is highly available
- Use coordination only if its unavoidable
  - Allows for Just-right-consistency
- Supports programmer with comprehensive interface
  - Abstract data-types (CRDTs) and transactions

# Conclusion: Part I

- Choice of consistency has consequences for system availability
- **CP systems** provide strong consistency, but require expensive infrastructure to provide high availability and introduce higher latencies
- **AP systems** opt for high availability, but provide weaker consistency guarantees and increase complexity in data management