On codifying viable and useful consistency levels to developers

Dharma Shukla, Engineering, Azure DocumentDB
(dharmas@microsoft.com, @dharmashukla)
HPTS 2015

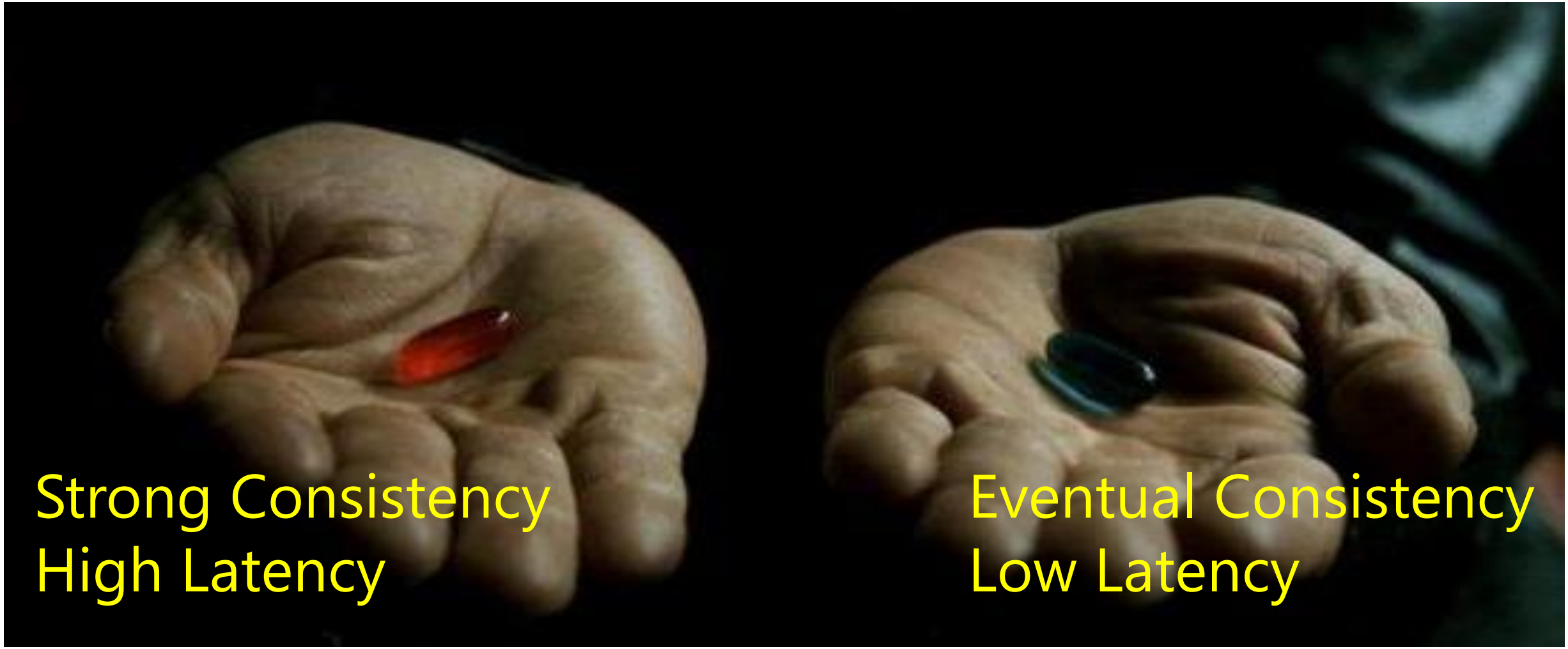
What is DocumentDB?

Fully managed, multi-tenant, geo-distributed document database service on Azure

- Born out of the needs of internal Microsoft applications; GA since April 2015; growing fast internally & externally
- Built from the ground up with resource governance
  - Performance isolation, OPEX efficiency
- Database engine built for JSON & JavaScript
  - Automatic indexing of JSON values and rich (SQL and JavaScript) query (*Schema Agnostic Indexing with Azure DocumentDB, VLDB 2015*)
  - JavaScript language integrated transactions and query directly inside the database engine
- Well defined consistency levels with predictable performance

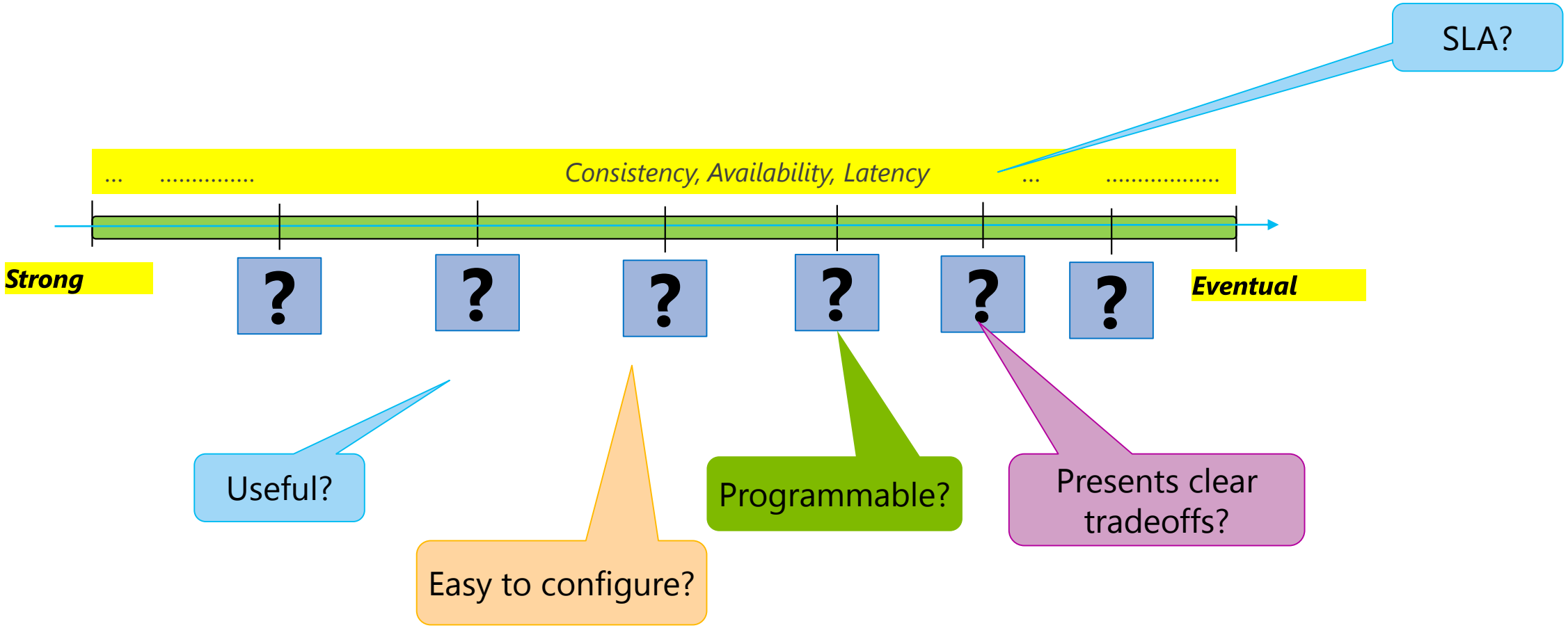
Too many to choose from

The state of cloud database services



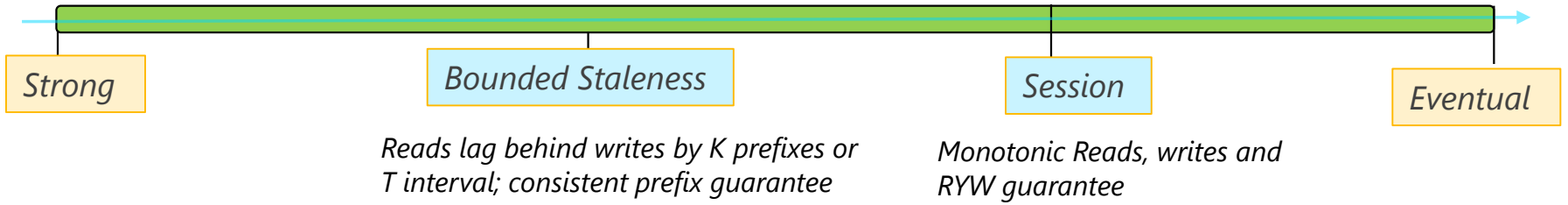
Inconsistent programming model for distributed applications

Engineering opportunity

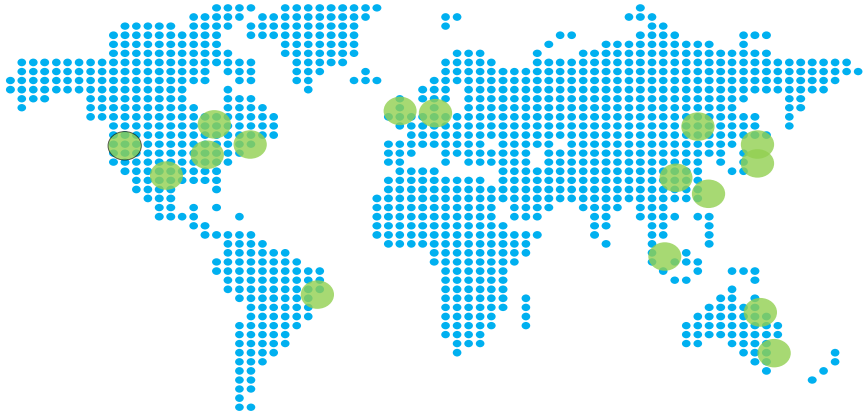


DocumentDB: Programming model

1 Choose "default" consistency level(*) for your database



2 Next, (subject to consistency level,) choose write region and read regions

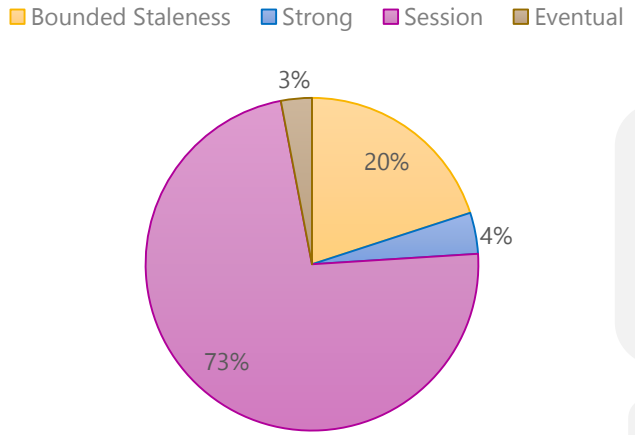


3 Finally, provision/scale throughput for each write/read region

(*)*Developers can weaken the consistency level on a per request basis.*

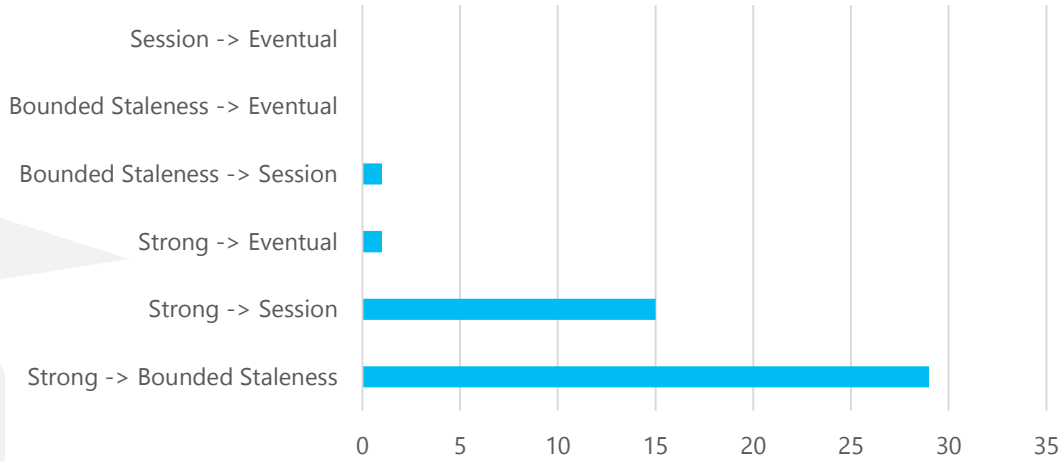
DocumentDB: Insights from production workloads

Default Consistency Level spread across 11 geo regions



- On average, only 2% of requests were overridden to a weaker consistency level
- Of those, downgrading to Session and Bounded Staleness popular

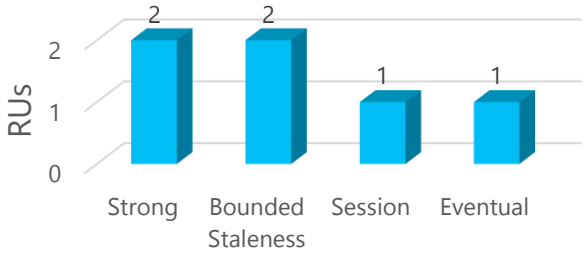
Spread among the requests which override default consistency



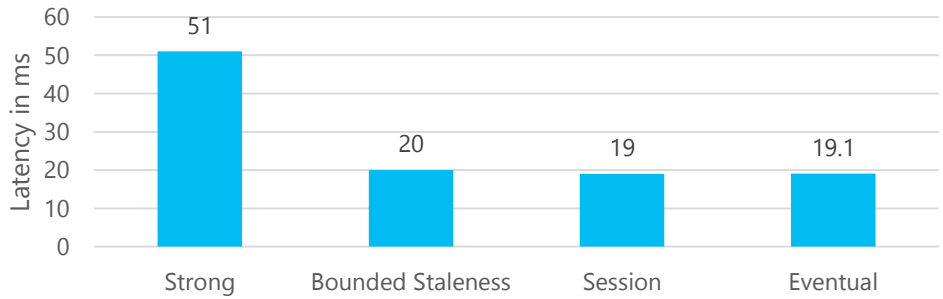
- Upper bound on request processing latency part of the availability SLA

Throughput provisioned in terms of "request units" (RUs)

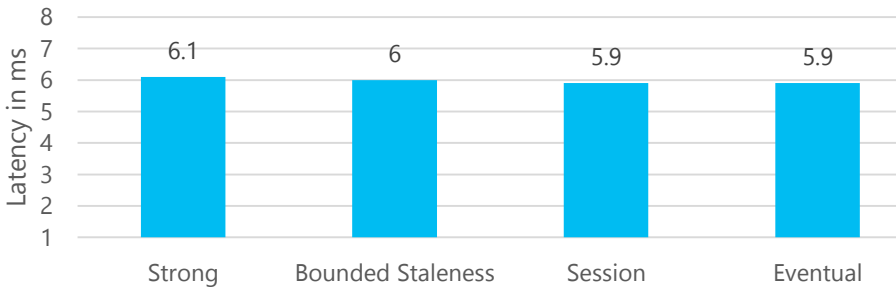
Read RUs (1 KB Doc)



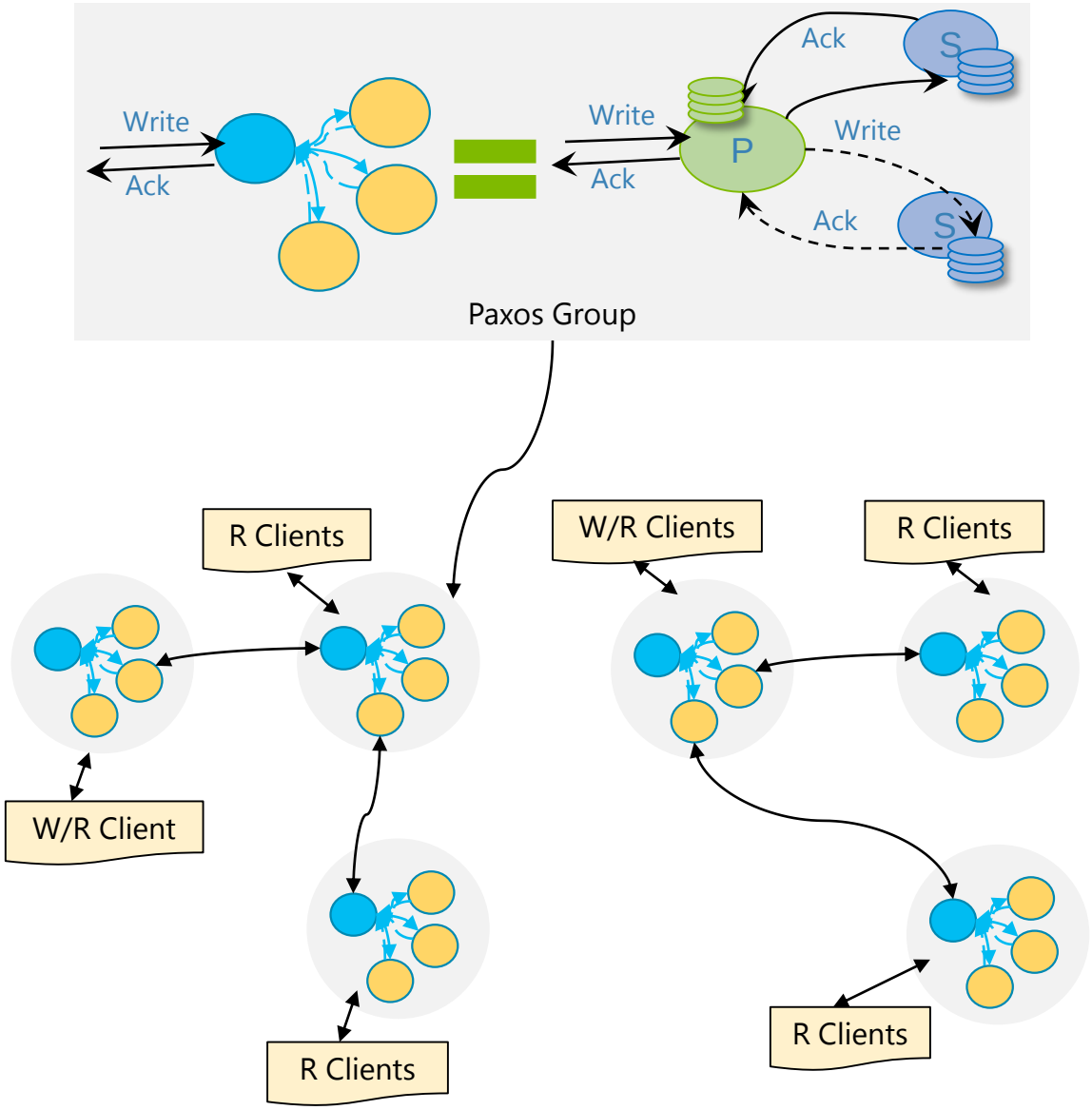
99% write latency 1KB fully/auto indexed documents (client observed – across various connectivity options)



99% read latency 1KB documents (client observed - across various connectivity options)



DocumentDB: Implementation



- Resource governed and composable coordination framework
- Self-adjustable quorums for better handling of successive failures
- Topology strategies for various consistency levels
- Dynamic placement and load balancing of replicas and groups driven by resource governance and provisioned throughput

Summary

Engineering opportunity lies between the two extremes

- Weaker consistency levels are valuable
- Useful, real-world use cases do exist
- A "consistent" programming model (and associated tradeoffs) are possible to codify

What are we up to?

- Operationalize new features via internal usage first
- Exploring "collection types" with CRDTs
- We intend to publish papers describing the system design and our production experiences

These are interesting times for distributed consistency! We'd love to collaborate!

@dharmashukla, dharmas@microsoft.com