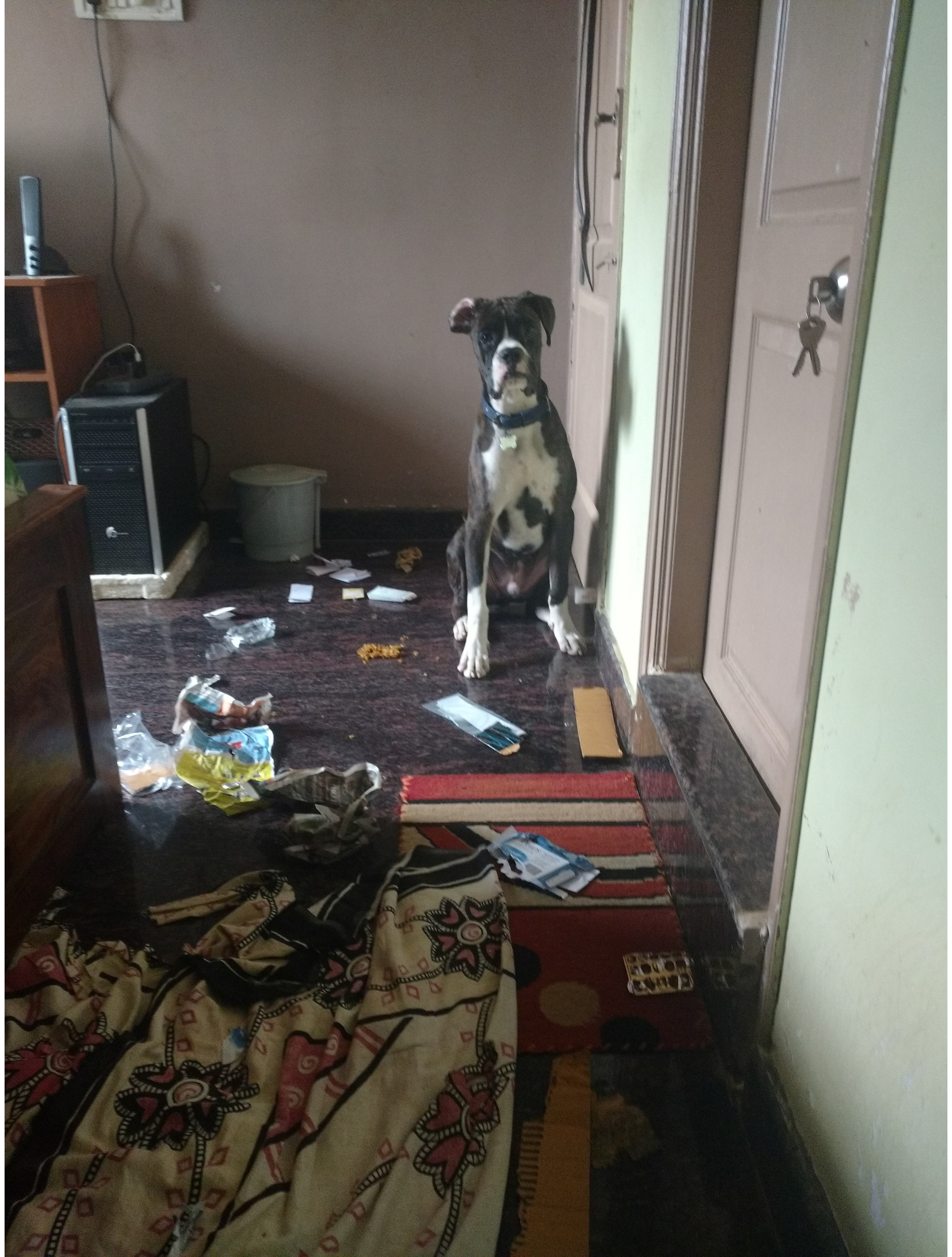
Scaling distributed Data Systems: A LinkedIn Case study

Sai Kiran Kanuri

Data Chef @LinkedIn

About me

- Currently at LinkedIn, messing with their data platforms for the past 20 months
- Previously bothered people at Walmart, Yahoo, Akamai & Standard Chartered Bank
- skanuri@linkedin.com
- https://www.linkedin.com/in/saikirankanuri/
- https://www.flickr.com/saikiranrgda



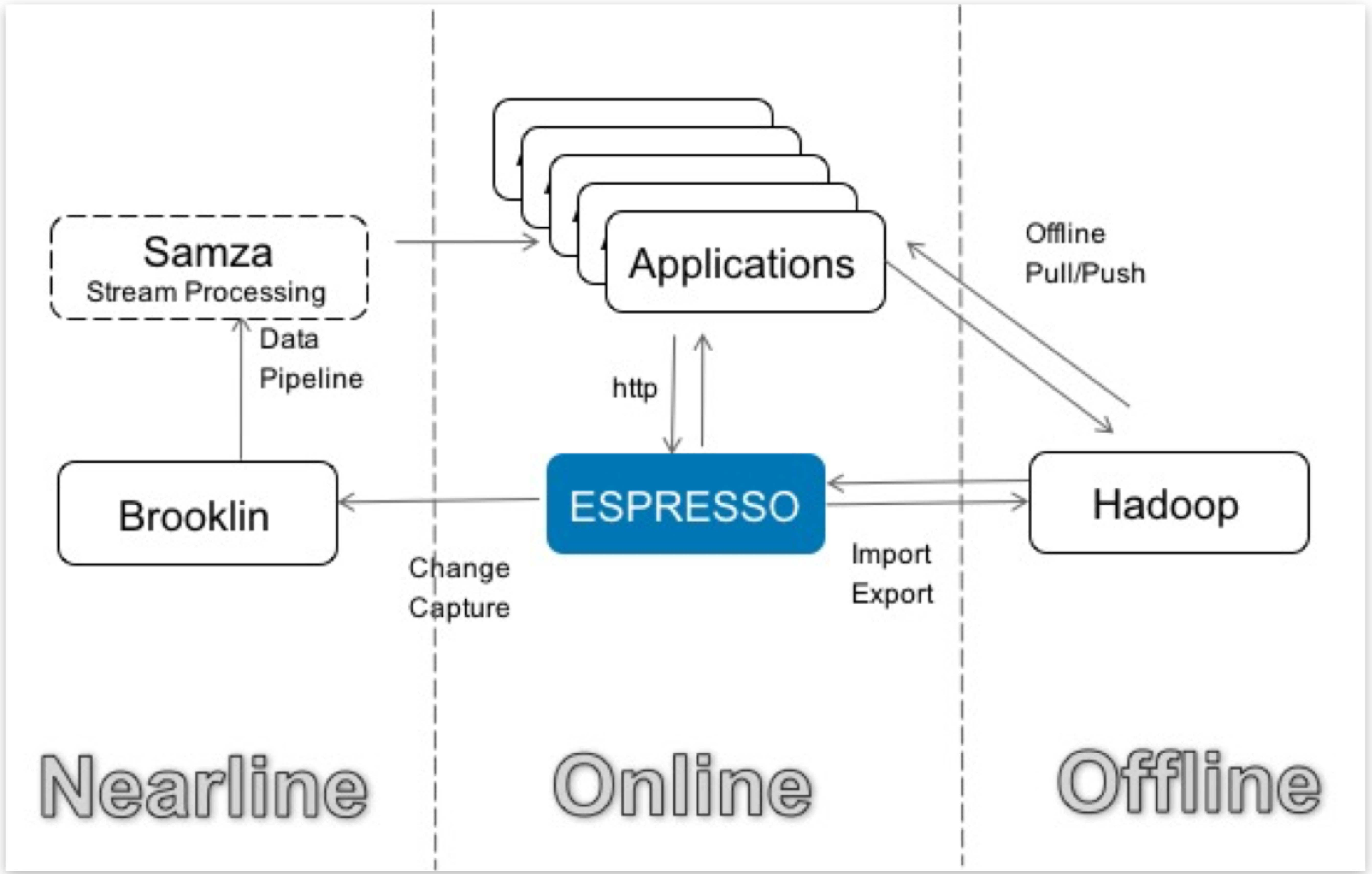
Today’s agenda

- Brief Intro to Espresso
- Challenges
- Approaches we taken
- Future Enhancements
- Additional Reading

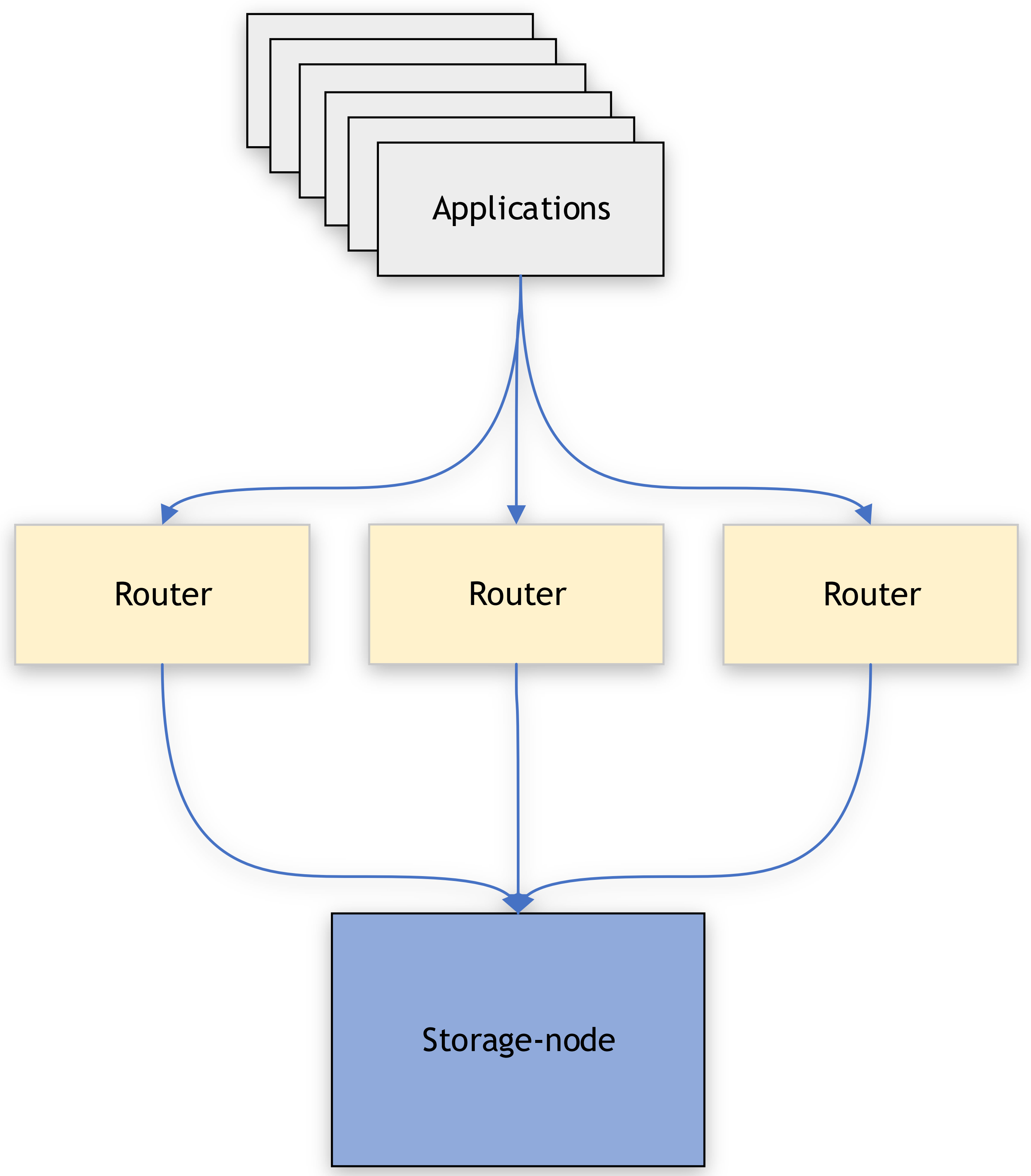
Espresso

- A online, distributed, fault tolerant NoSQL DB
- Multi Master cross colo support
- Bridges gap between RDBMS & k-v stores
- Hosts some of the most heavily accessed and valuable datasets at LinkedIn
- Replicates data for global availability & geo-locality.

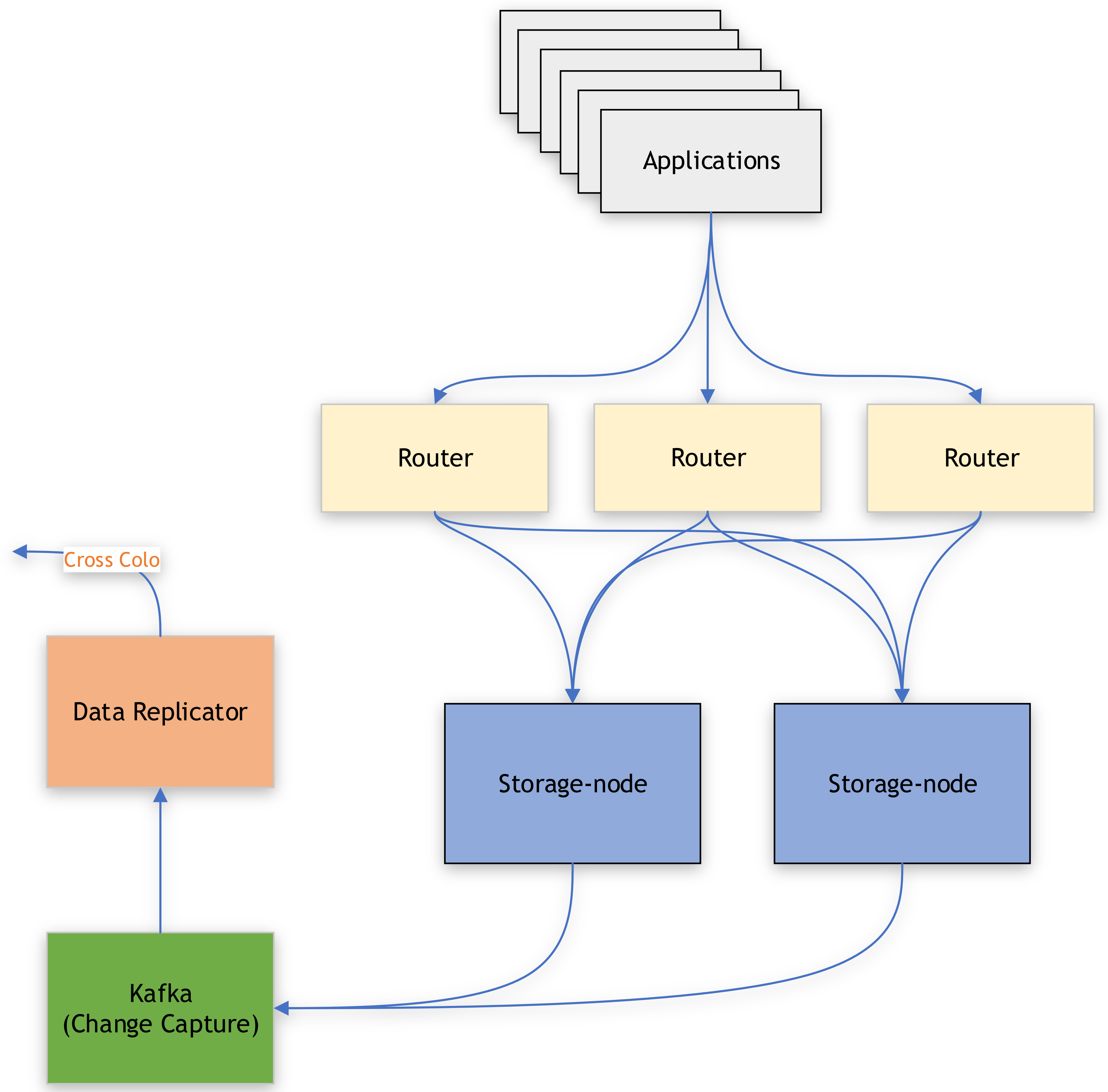
Espresso in LinkedIn



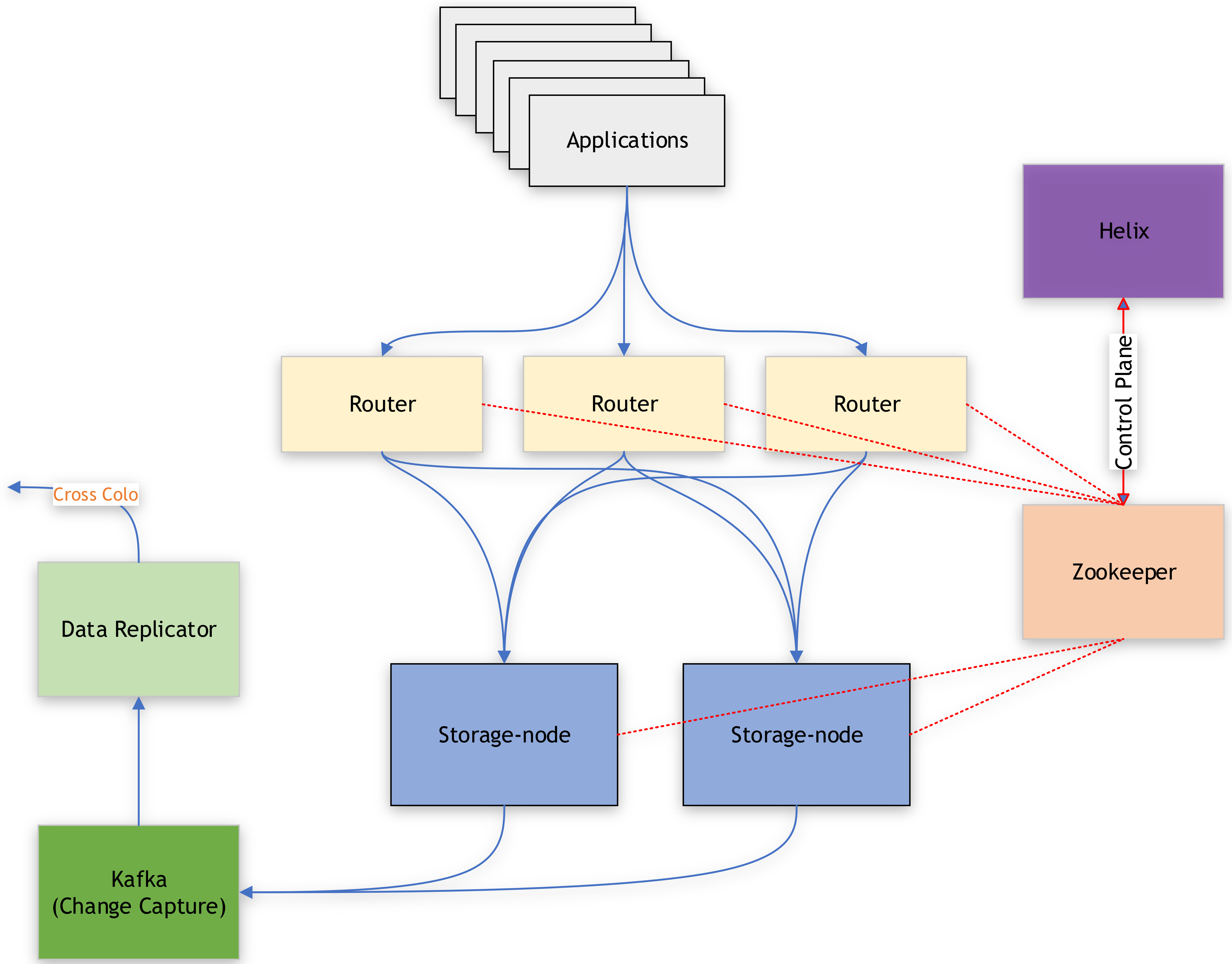
Espresso Architecture



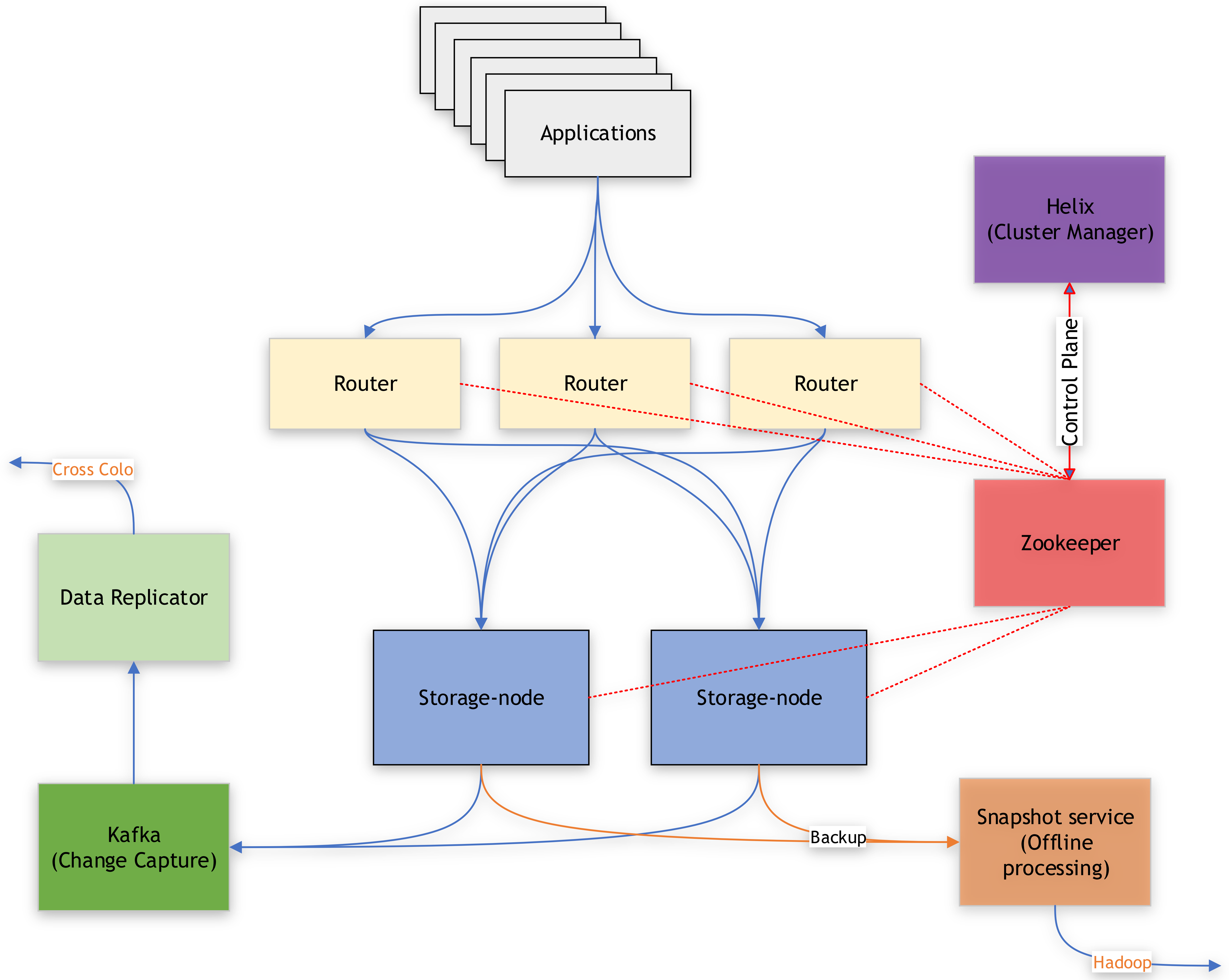
Espresso Architecture



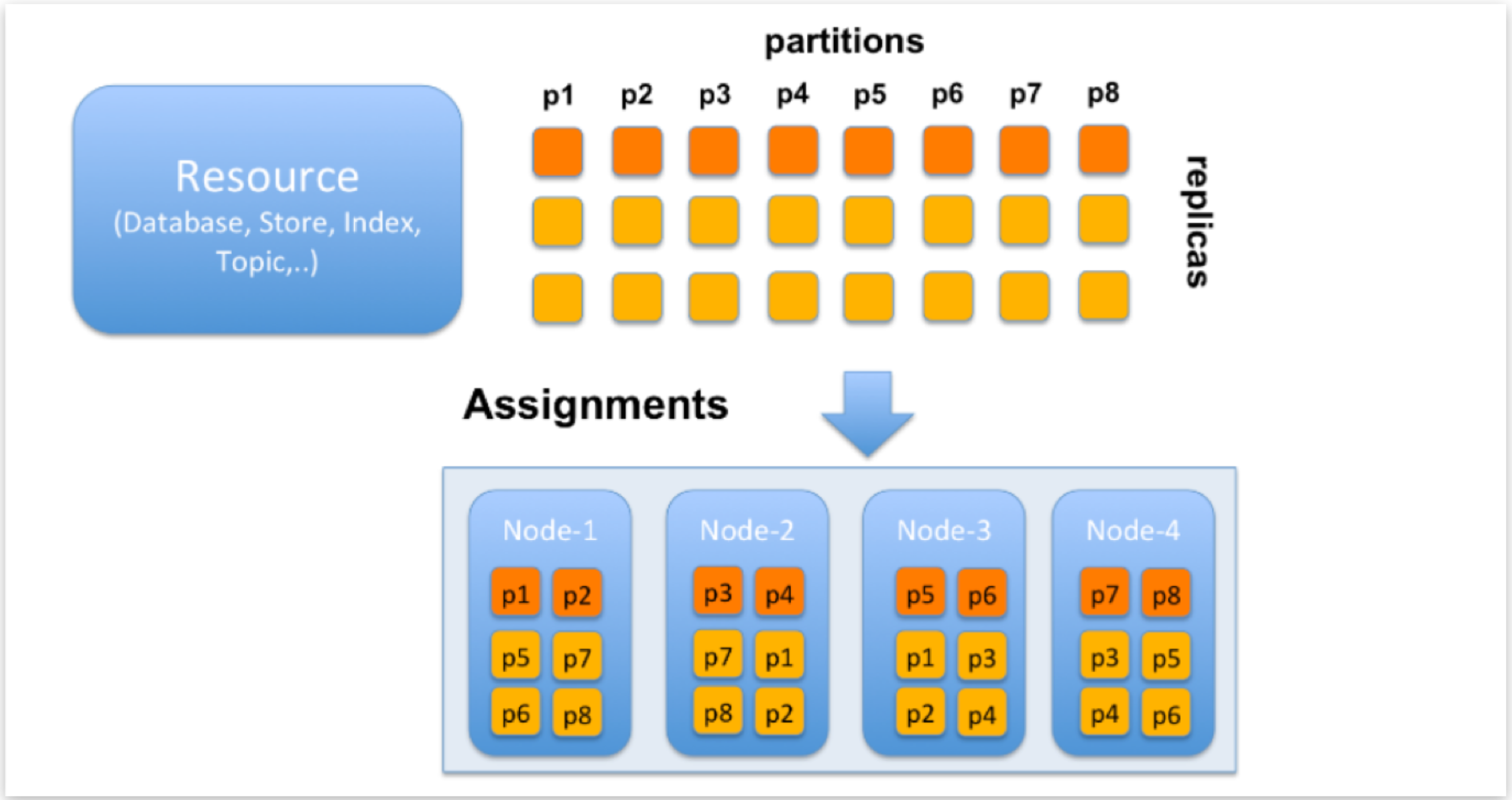
Espresso Architecture



Espresso Architecture



Espresso Storage Node



What is the biggest Challenge?

Data!

Challenges

- Interactive nature of user requests
- Low latency requirements
- Highly Volatile nature of social media requests

Challenges

Fault Tolerance

Multi-tenancy

Challenges in Fault Tolerance

- Nodes fail all the time
- Failover time
- Data replication
- Partition movement during failures

Approach to Fault Tolerance

- Minimize resource sizes
- Distribute data movement across cluster
- Minimize data movement
- Throttle data migration
- Minimize latency in control events

Approach to Fault Tolerance

- Utilize nocturnal traffic patterns for system maintenances
- Load balancing with error back-off
- Revisit read after write consistency requirements for different clients
- Enable client re-tries based on response codes

Challenges in Multi-Tenancy

- Security
- Data growth
- Hot partitions
- Service Discovery

Approach to Multi-Tenancy

- Ensure tenant isolation
- Quotas
- Transparently migrate data from one cluster to another
- Be able to re-partition existing data
- Data movement should be transparent to customer.
- Have a schema review process in place.

Future Changes

- Weighed nodes & partitions
- Dynamic data re-partitioning
- Automatic cross cluster data rebalancer
- Improved MTTR with P2P communication

Additional Reading

- Espresso - https://engineering.linkedin.com/espresso/introducing-espresso-linkedins-hot-new-distributed-document-stor
- R2D2 - https://github.com/linkedin/rest.li
- Brooklin - https://engineering.linkedin.com/blog/2017/10/streaming-data-pipelines-with-brooklin
- Helix - https://engineering.linkedin.com/apache-helix/apache-helix-framework-distributed-system-development
- Helix - https://engineering.linkedin.com/blog/2017/07/powering-helix_s-auto-rebalancer-with-topology-aware-partition-p
- Crush - https://ceph.com/wp-content/uploads/2016/08/weil-crush-sc06.pdf

Acknowledgements

- oakleyoriginals for funny Trash Dogs picture.
- The SCADS Director: Scaling a Distributed Storage System Under Stringent Performance Requirement paper by Beth Trushkowsky, Peter Bod´ık, Armando Fox, Michael J. Franklin, Michael I. Jordan, David A. Patterson
- Too Big to Fail by Kode Vicious

Questions

Thank You

+