

ACM SIGCOMM 2017 Workshop on Mobile Edge Communications
(MECOMM'2017), Los Angeles



Managing Data in Computational Edge Clouds

Nitinder Mohan* Pengyuan Zhou* Keerthana Govindaraj[△] Jussi Kangasharju*

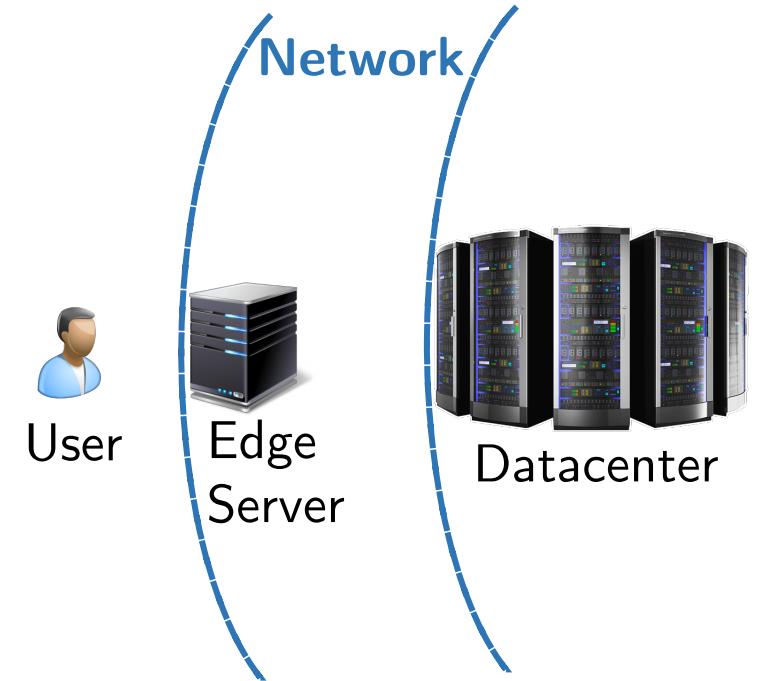
*University of Helsinki [△]Robert Bosch GmbH

*{firstname.lastname@helsinki.fi} [△]{keerthana.govindaraj@de.bosch.com}

Rise of Edge Clouds

Edge Cloud: Small-scale server(s)
deployed at network edge to compute user
data

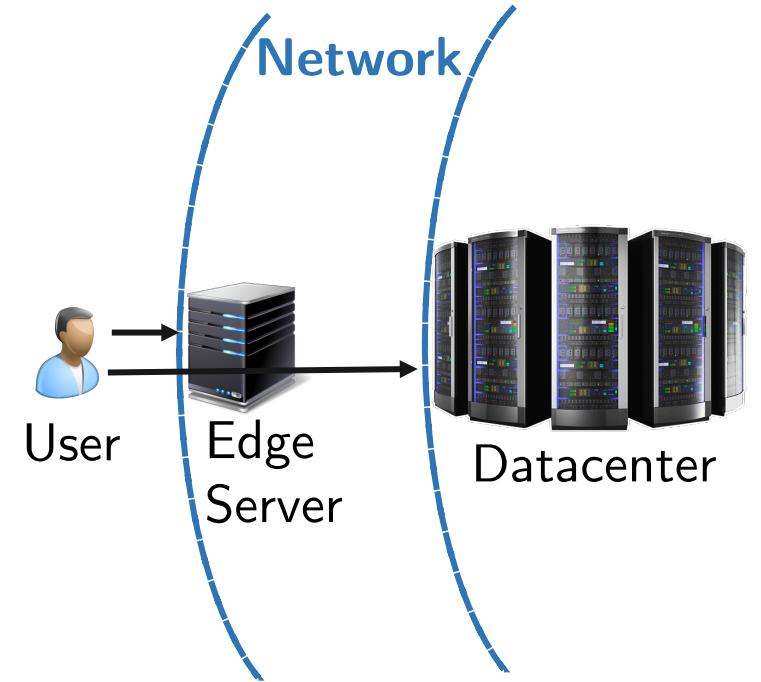
Motivation:



Rise of Edge Clouds

Edge Cloud: Small-scale server(s)
deployed at network edge to compute user
data

Motivation:
✓ Decreased latency and network traffic

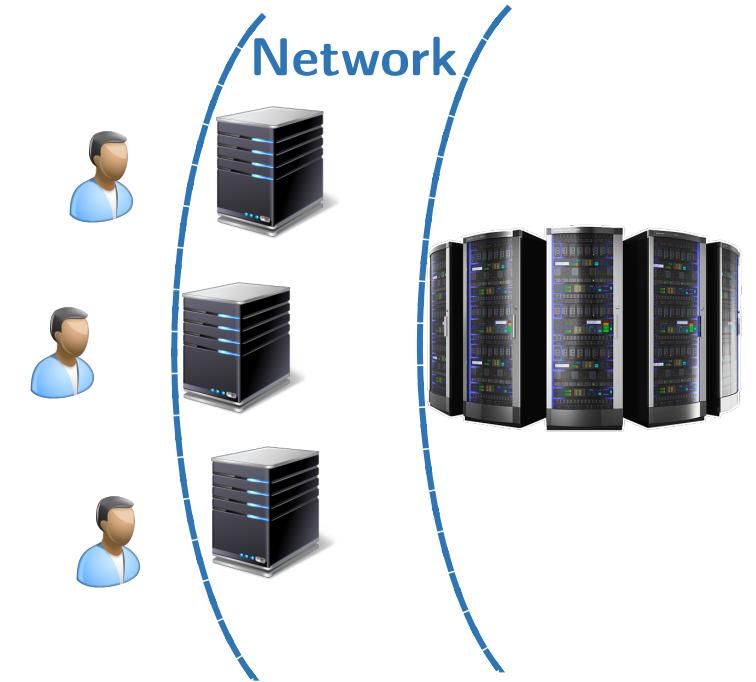


Rise of Edge Clouds

Edge Cloud: Small-scale server(s) deployed at network edge to compute user data

Motivation:

- ✓ Decreased latency and network traffic
- ✓ Computing data of local relevance



Industry 4.0 on Edge

Industry 4.0 proposes *automation in manufacturing*

1. Robots such as **Bosch APAS** can automate tasks and work along side humans while ensuring safety
2. **IoT sensor data** is used in augmented reality for collaborative support and manufacturing

Requirement: Time-critical and fast data processing



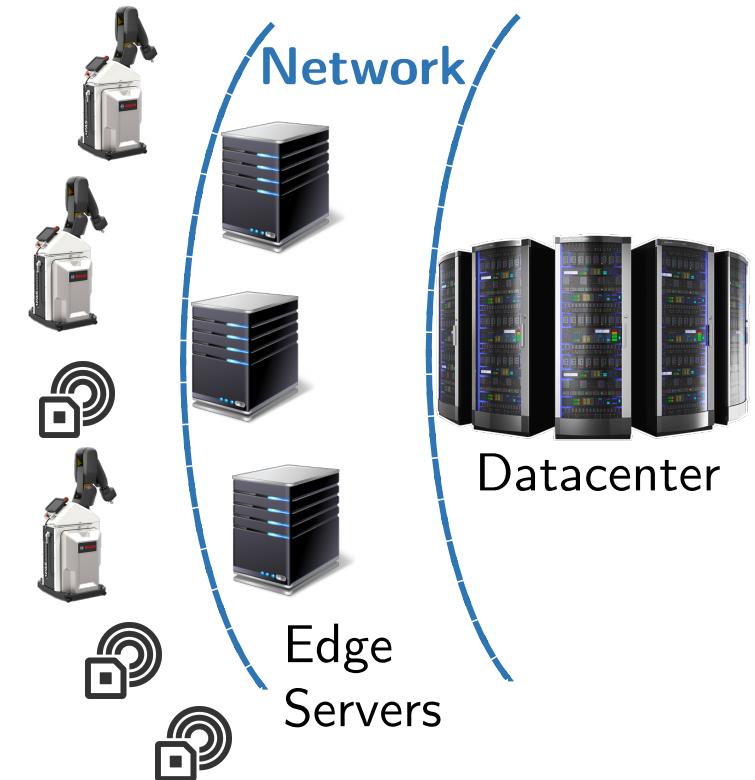
Industry 4.0 on Edge

Industry 4.0 proposes *automation in manufacturing*

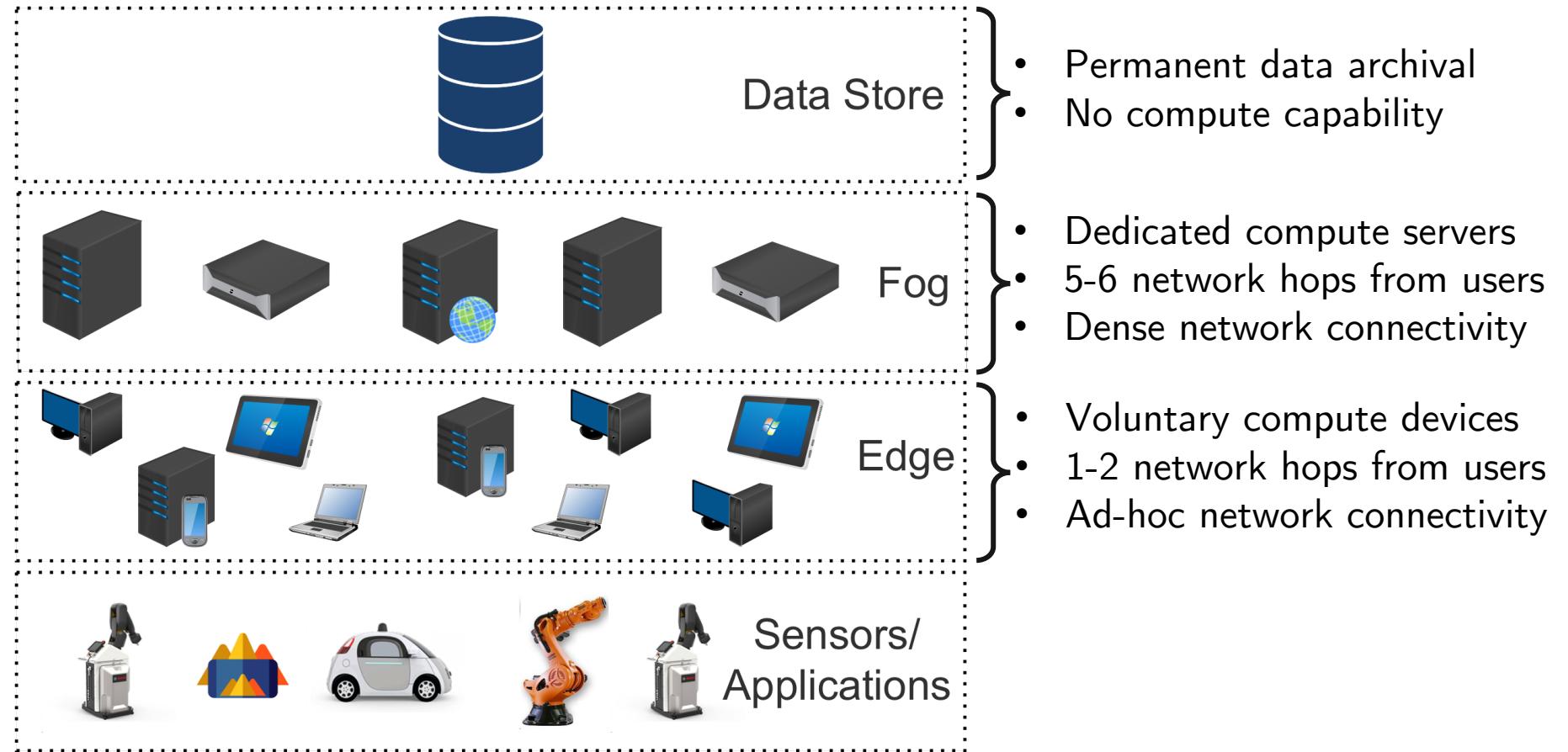
1. Robots such as **Bosch APAS** can automate tasks and work along side humans while ensuring safety
2. **IoT sensor data** is used in augmented reality for collaborative support and manufacturing

Requirement: Time-critical and fast data processing

Solution: Implement industrial Edge clouds

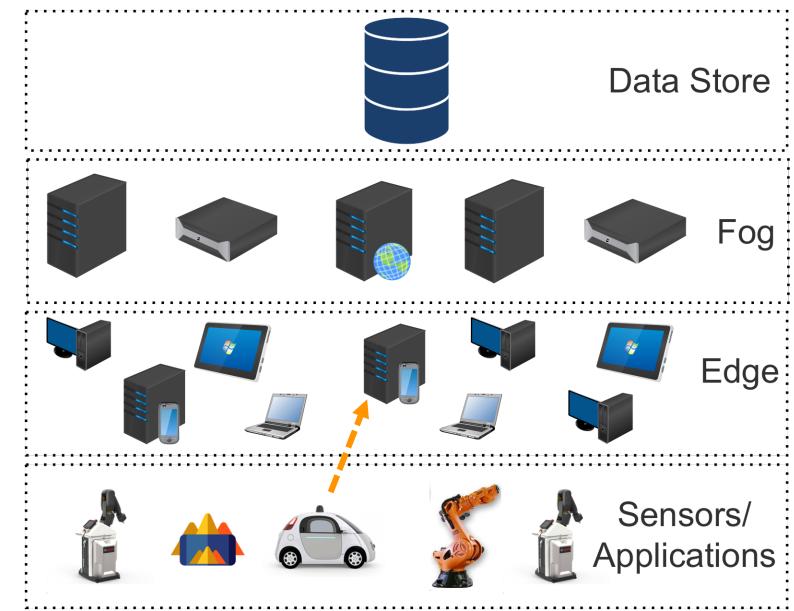


Edge-Fog Cloud



Nitinder Mohan and Jussi Kangasharju. "Edge-fog cloud: A distributed cloud for internet of things computations." *Cloudification of the Internet of Things (CIoT)*. IEEE, 2016.

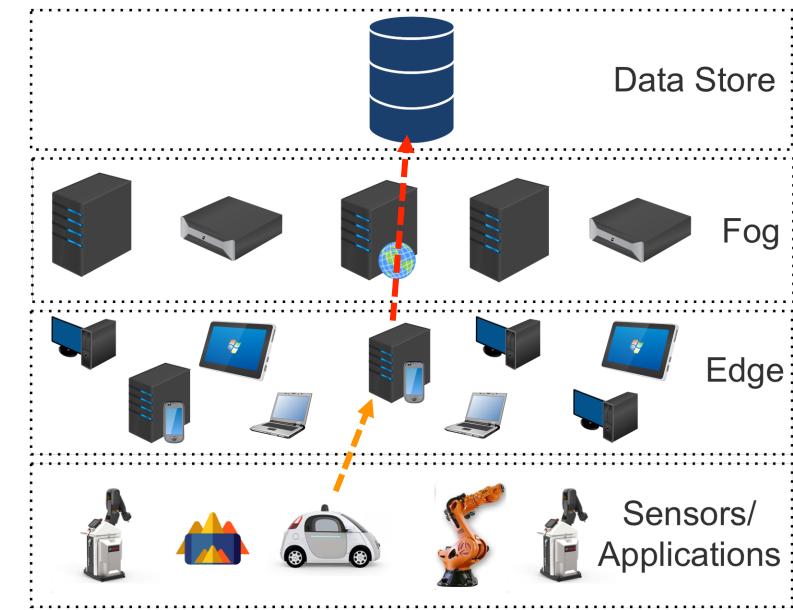
Data Availability in Edge Cache



Query: Give me a route from point A to B without road disruptions

Data Availability in Edge Cache

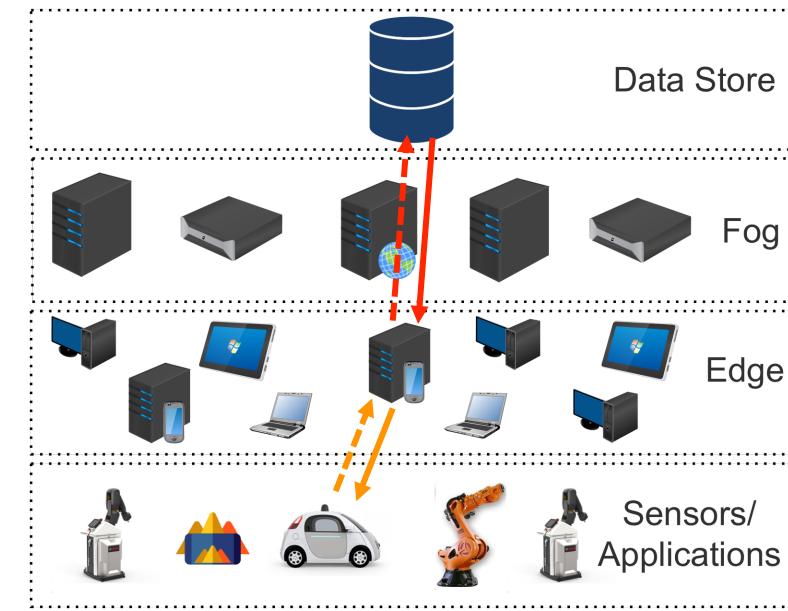
- Edge cloud proposals presume that the required data for processing is pre-available in Edge servers cache
- However, Edge server may request Data Store for required data



Data Request: Any construction/
accidents on the map?

Data Availability in Edge Cache

- Edge cloud proposals presume that the required data for processing is pre-available in Edge servers cache
- However, Edge server may request Data Store for required data
- The computation is delayed by a **round-trip-time** to central Data Store
- **Reduces benefits of employing Edge clouds!**



Data Request: Any construction/
accidents on the map?

Requirement: Caching strategy which predicts and stores required data before computation

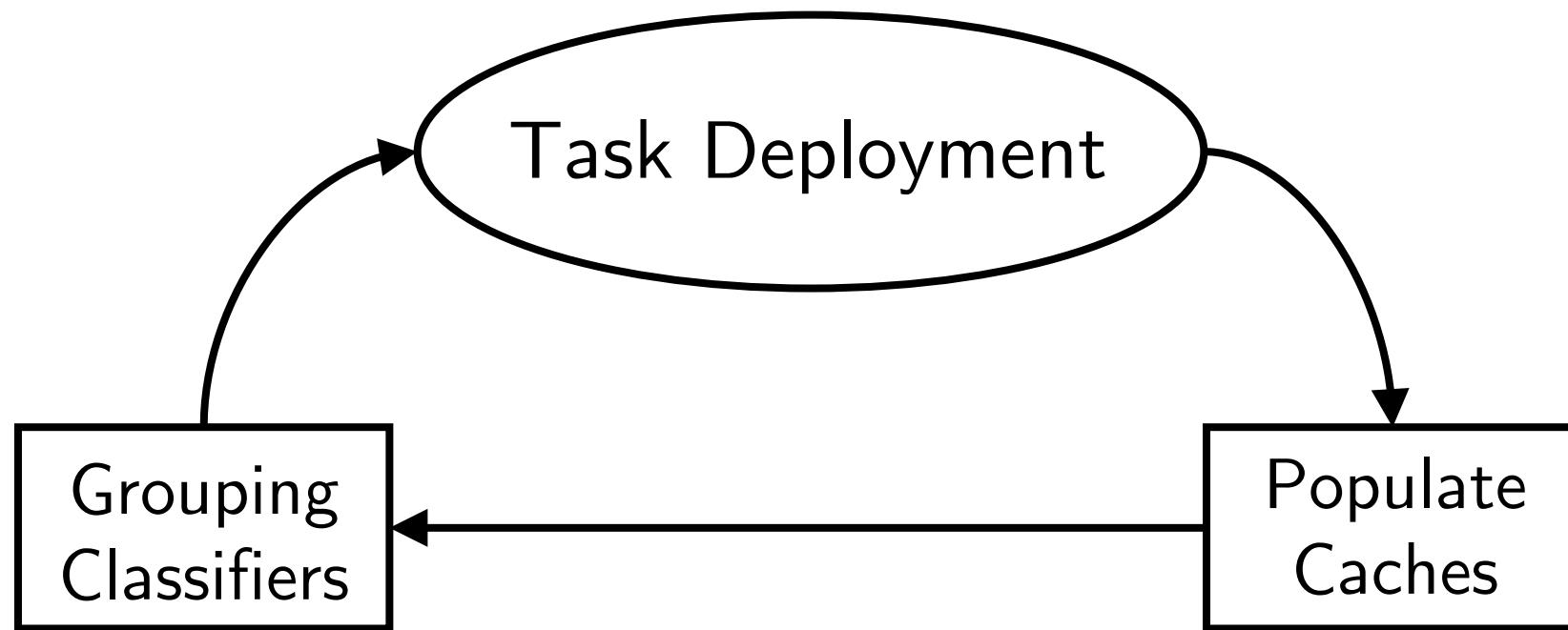
Edge Cloud Cache Grouping

Edge cloud caching mechanism

- Group edge resources into cache groups based on their locally cached data
- Predict required data for computation based on cache grouping classification
- Provide coherence on multiple copies of shared data throughout a cache group

Resource Cache Grouping

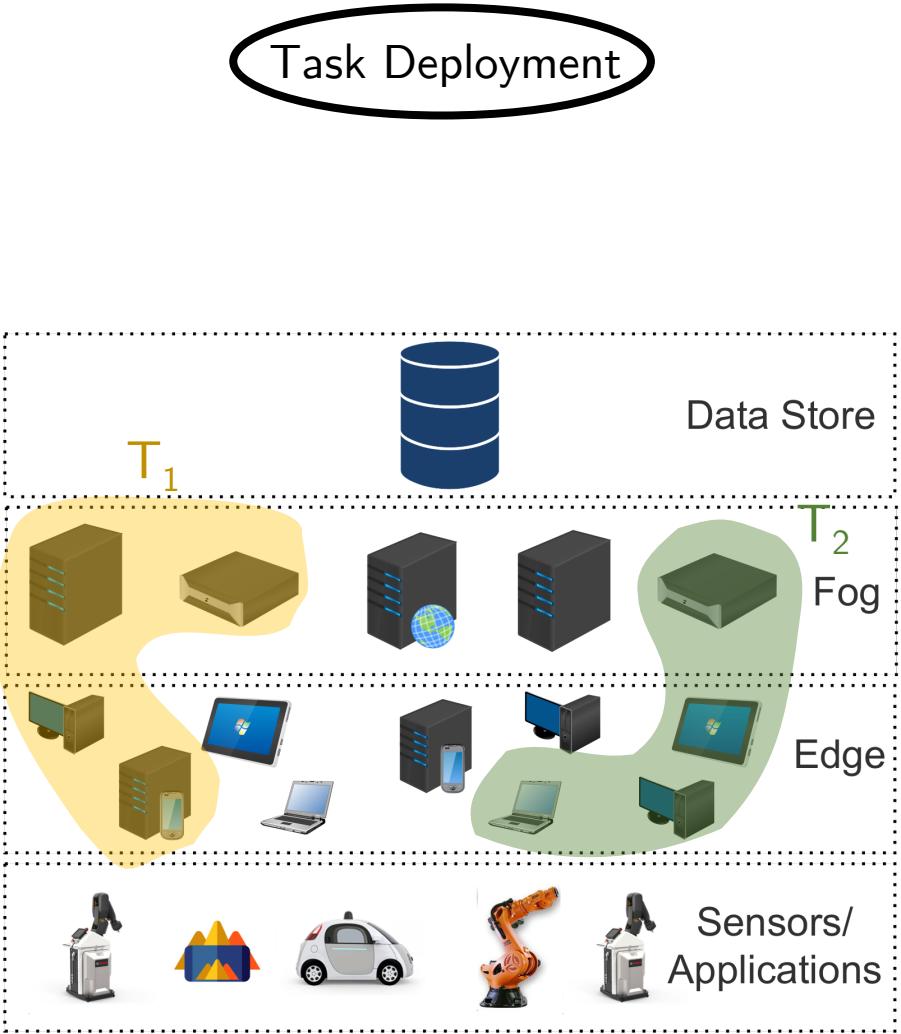
Grouping Algorithm



Grouping Algorithm

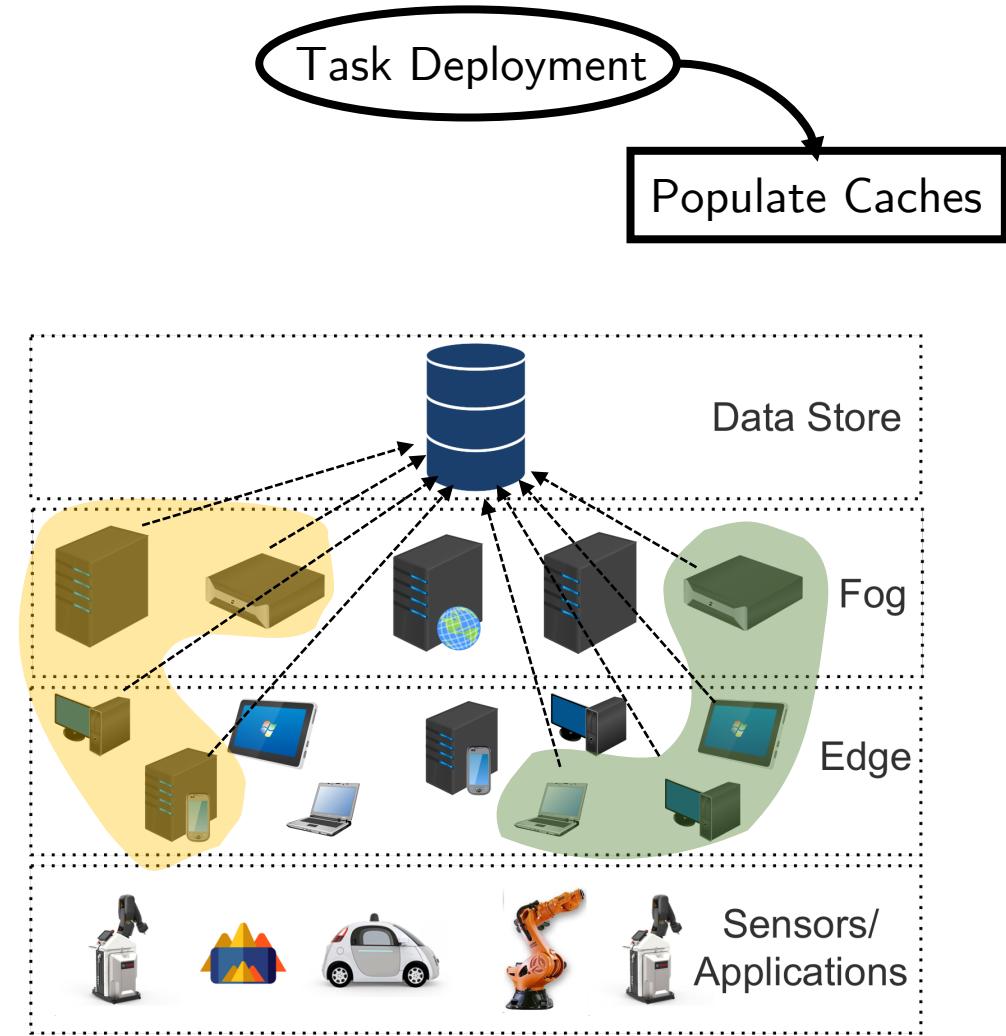
Deployment: Select $\{RC\}_i$ resources for computing a task T_c

1. Algorithms such as LPCF* selects a subset from available resources which achieves least processing and network cost for collaboration.
2. Several deployments for different tasks can occur at same time.



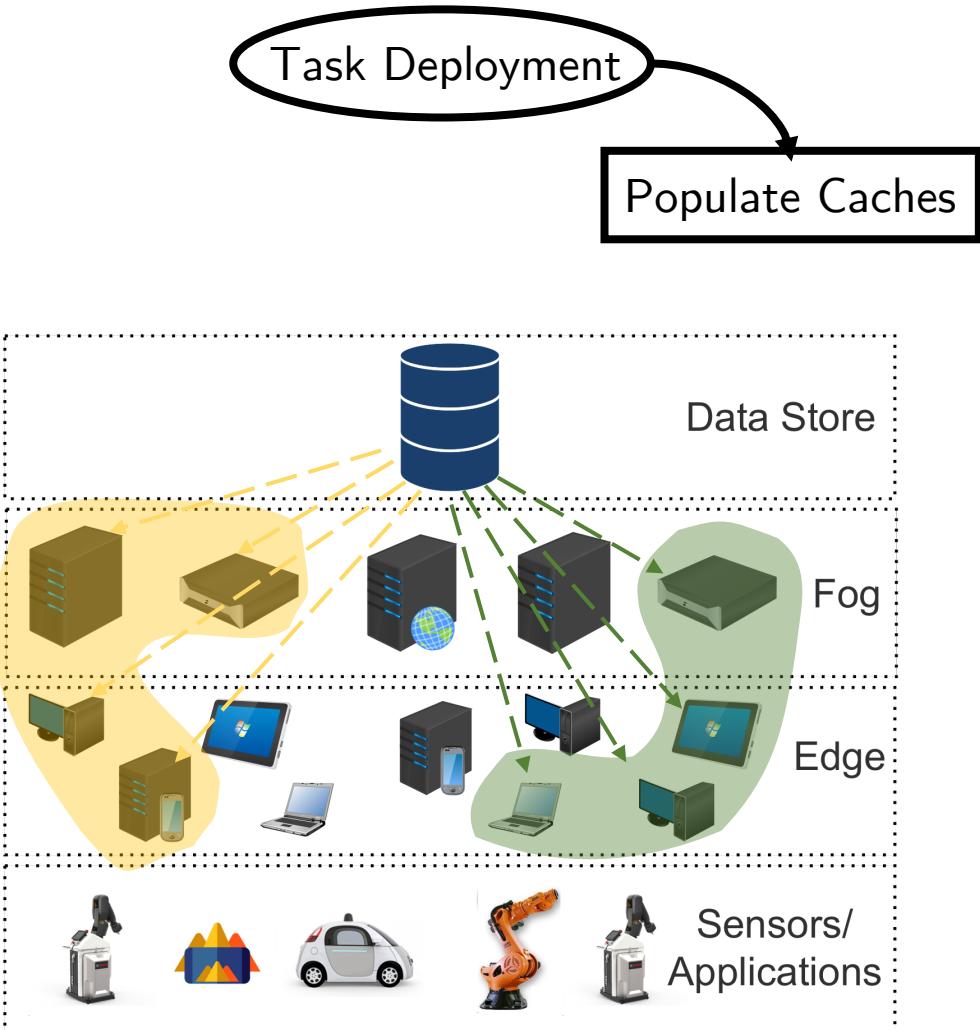
Grouping Algorithm

1. At $t=0$, $\{RC\}_i$ does not have data required for computation and it requests the required data from Data Store
2. The Data Store sends the required data to requesting resources **populating** their caches



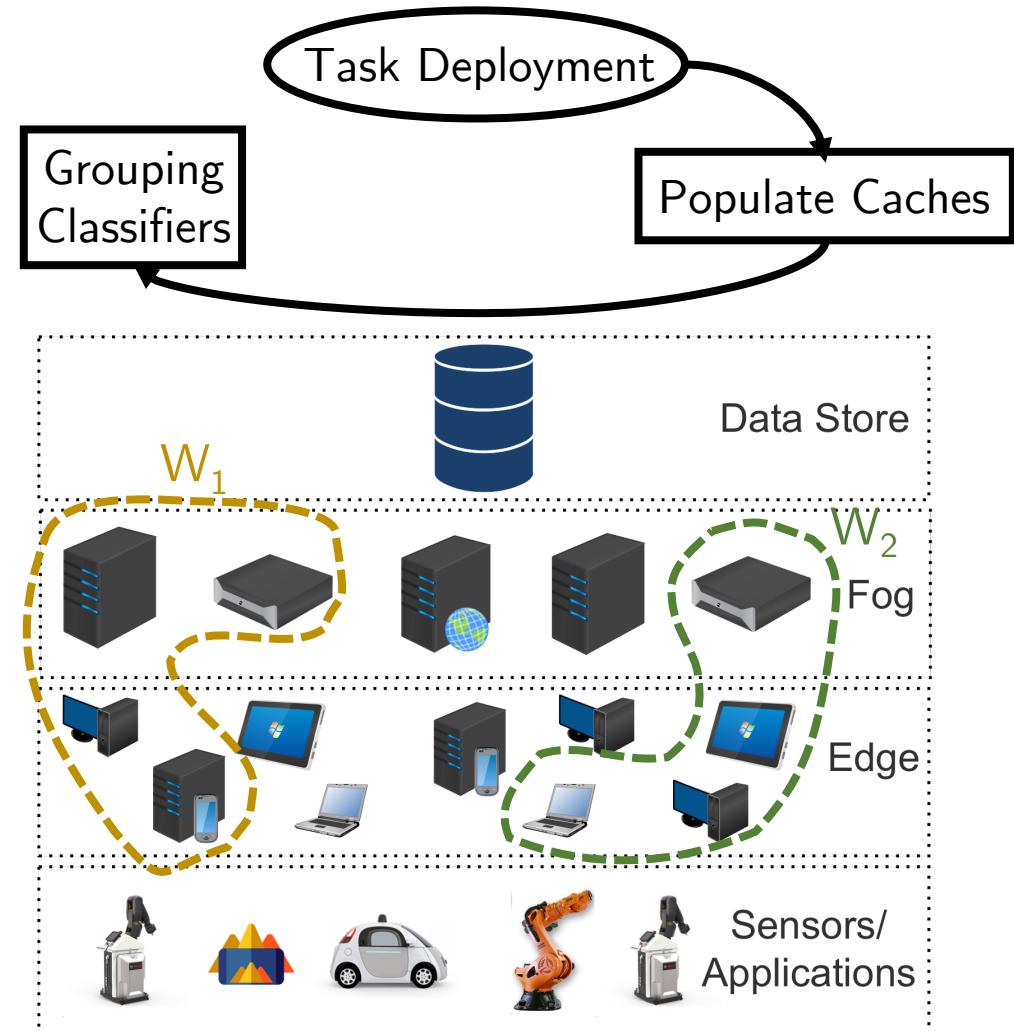
Grouping Algorithm

1. At $t=0$, $\{RC\}_i$ does not have data required for computation and it requests the required data from Data Store
2. The Data Store sends the required data to requesting resources **populating** their caches



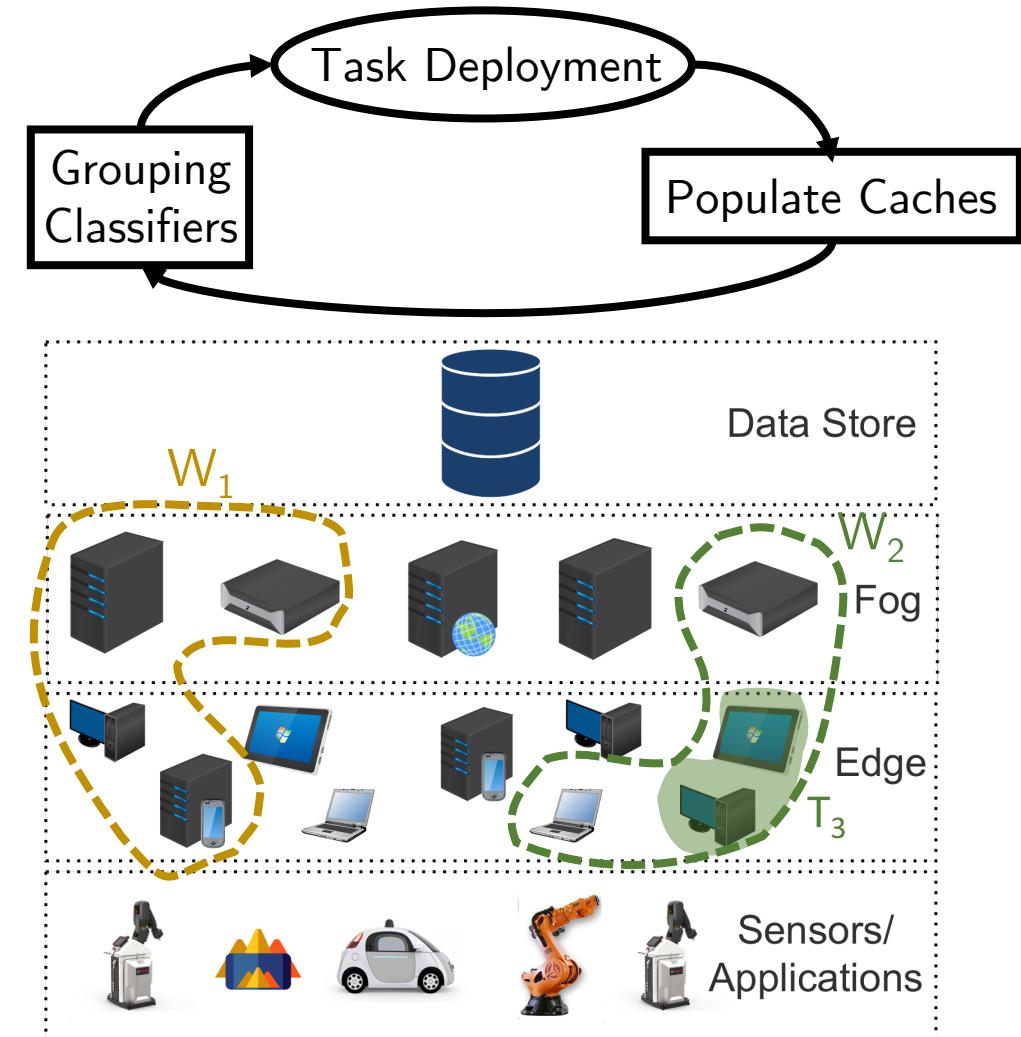
Grouping Algorithm

1. The task T_c is classified in a workload W_k where W_k is a set of similar tasks
2. The resources $\{RC\}$, caching data of workload W_k are grouped together as cache group CG_k



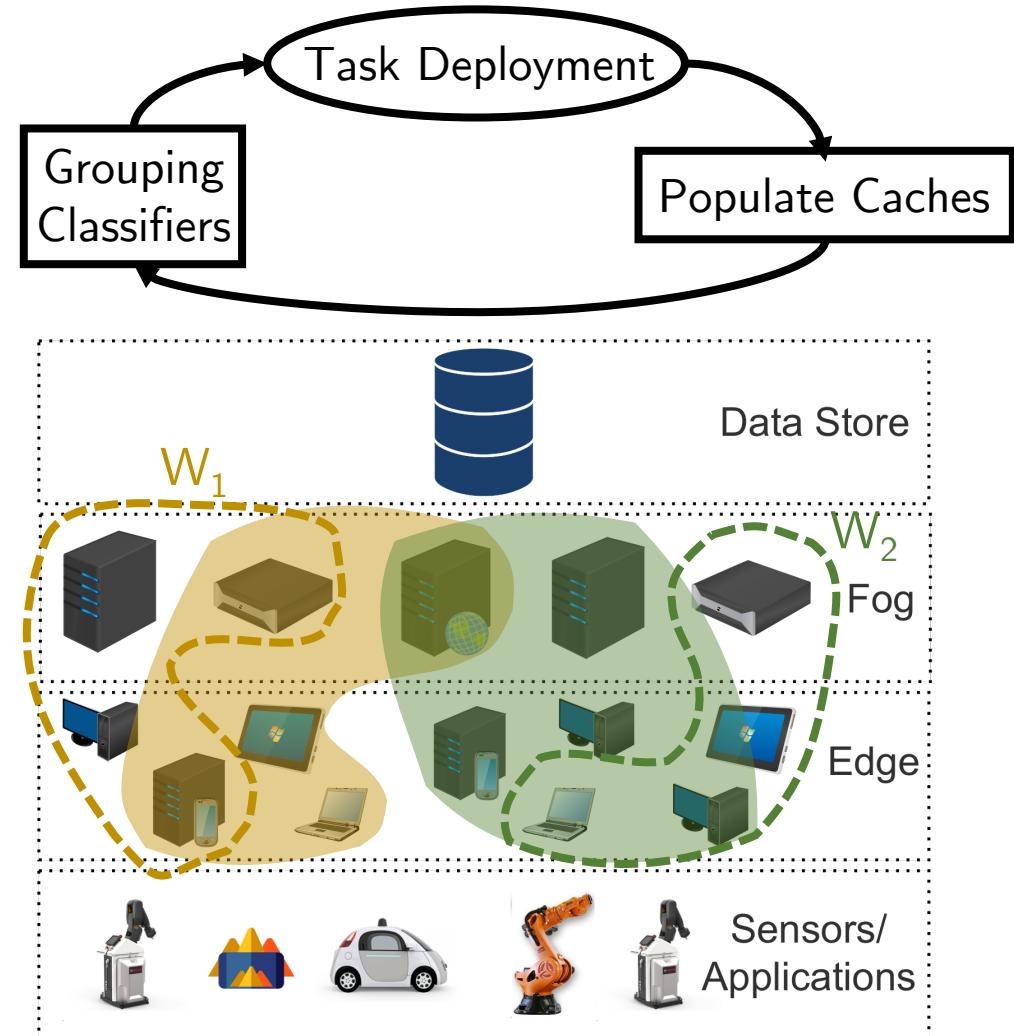
Grouping Algorithm

1. The deployment algorithm prefers $CG_k \in W_k$ for task $T_i \in W_k$



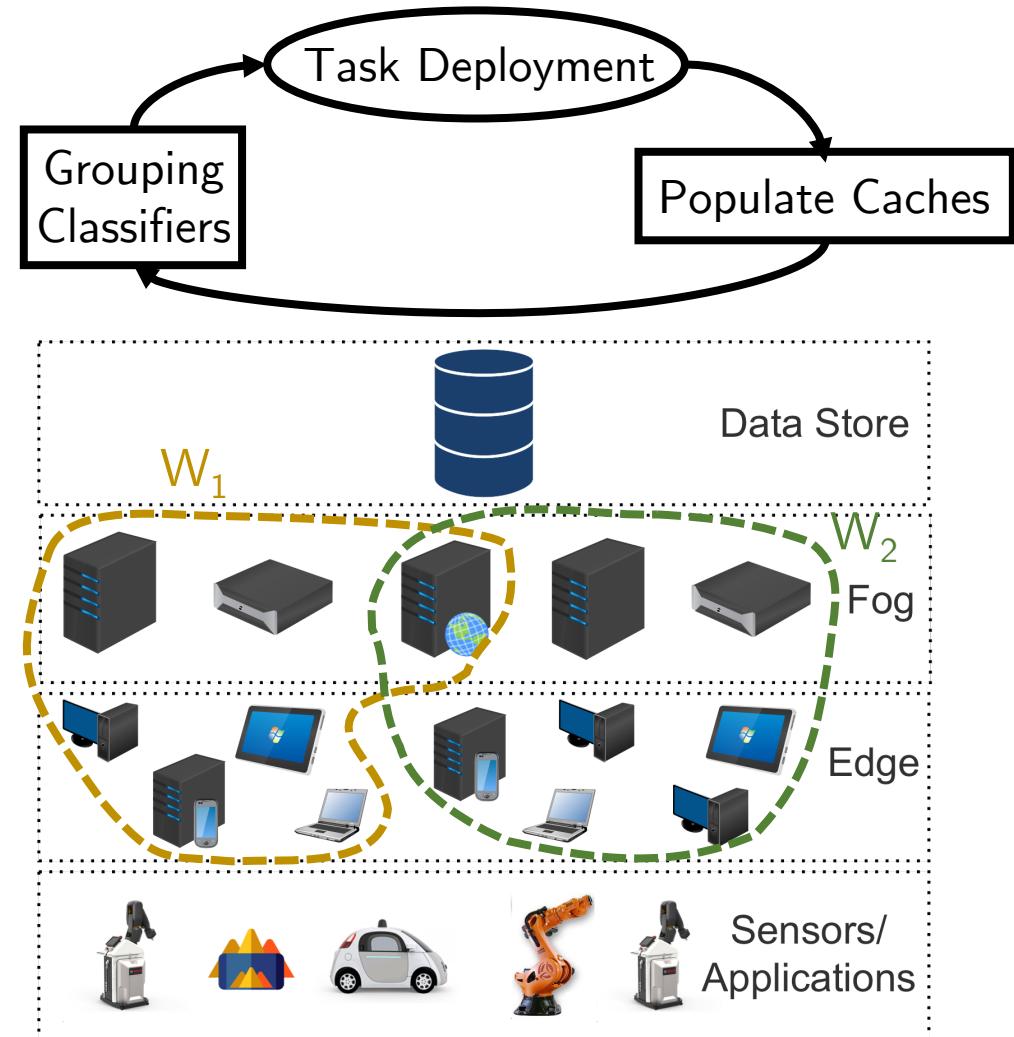
Grouping Algorithm

1. The deployment algorithm prefers $CG_k \in W_k$ for task $T_i \in W_k$
2. If the resources in group are not appropriate for deployment, the algorithm can select other available resources



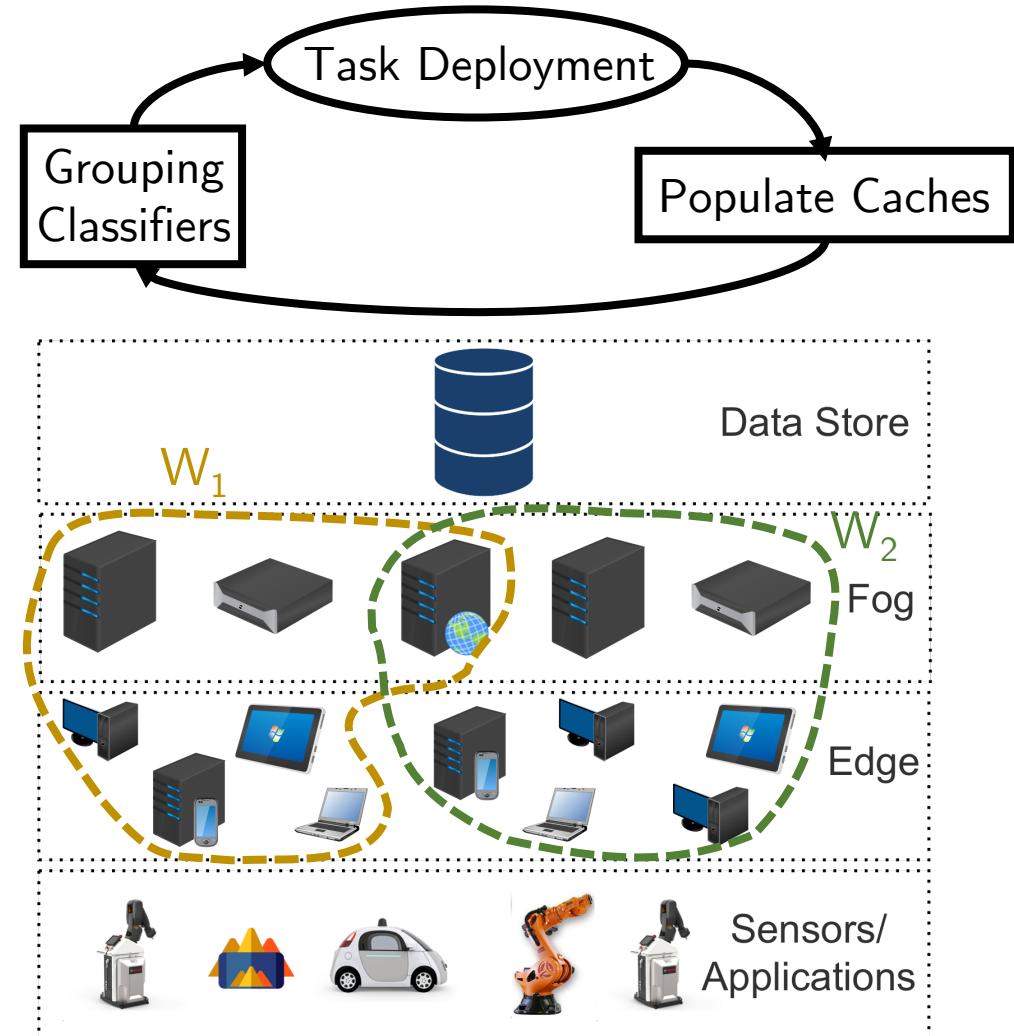
Grouping Algorithm

1. The deployment algorithm prefers $CG_k \in W_k$ for task $T_i \in W_k$
2. If the resources in group are not appropriate for deployment, the algorithm can select other available resources
3. These resources then join their respective cache group



Grouping Algorithm

1. The deployment algorithm prefers $CG_k \in W_k$ for task $T_i \in W_k$
2. If the resources in group are not appropriate for deployment, the algorithm can select other available resources
3. These resources then join their respective cache group
4. A resource can be a part of more than one group



Group Communication

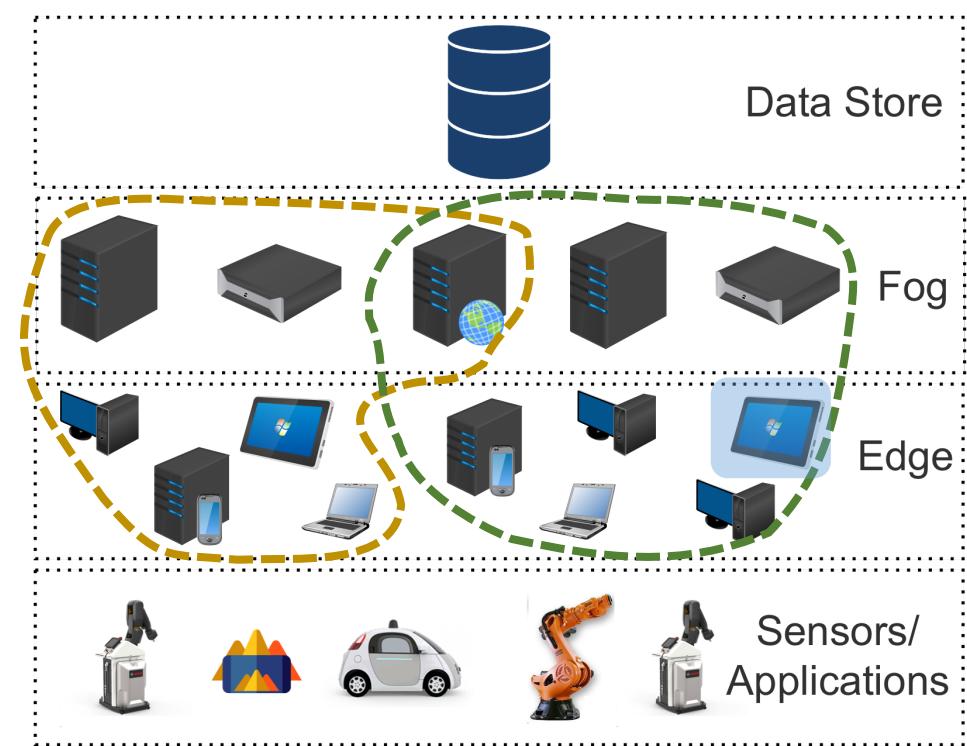
Requirements

1. Reduce *unnecessary* network traffic
2. Ensure *consistent computation* by avoiding stale data copies

Resource Roles

1. Group Member

- Resource (RC_x) part of a cache group $\{CG\}_i$
- The group member stores only data of its group workload



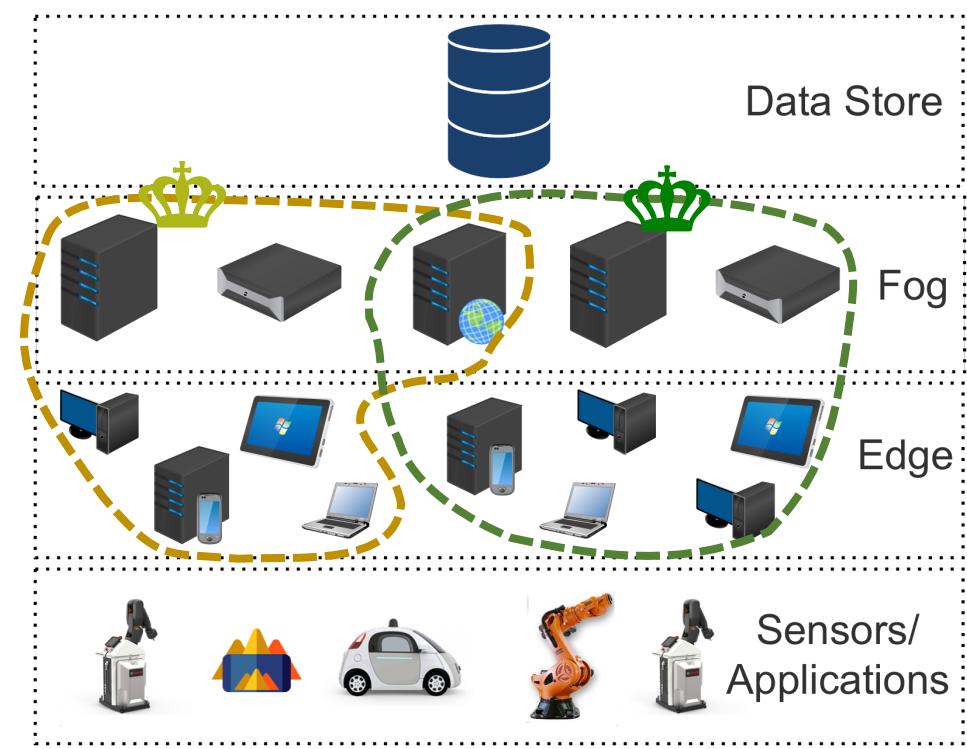
Resource Roles

1. Group Member

- Resource (RC_x) part of a cache group $\{CG\}_i$
- The group member stores only data of its group workload

2. Group Leader

- Representative of a group
- Responsible for maintaining coherent data within a group
- Has consistent connectivity with group members
- Elected via distributed election algorithm



Communication Flow

The communication model is strictly **pull-based** to limit number of messages in the system

The model operates over following **assumptions**:

1. The resources upload their updated data copy to Data Store after every successful computation
2. Data upload is in-parallel to ongoing local computation and does not impact the computation completion time
3. Data Store is persistent and immune to failures
4. Messages in the system are not lost

Cache Data Structures

1. Group member table

Maps locally cached data with group leader

Data Name	Leader Address	Tag
-----------	----------------	-----

2. Group leader table

Ensures that group leader has consistent state and address of resource caching data

Data Name	Resource Address	Tag	Timestamp
-----------	------------------	-----	-----------

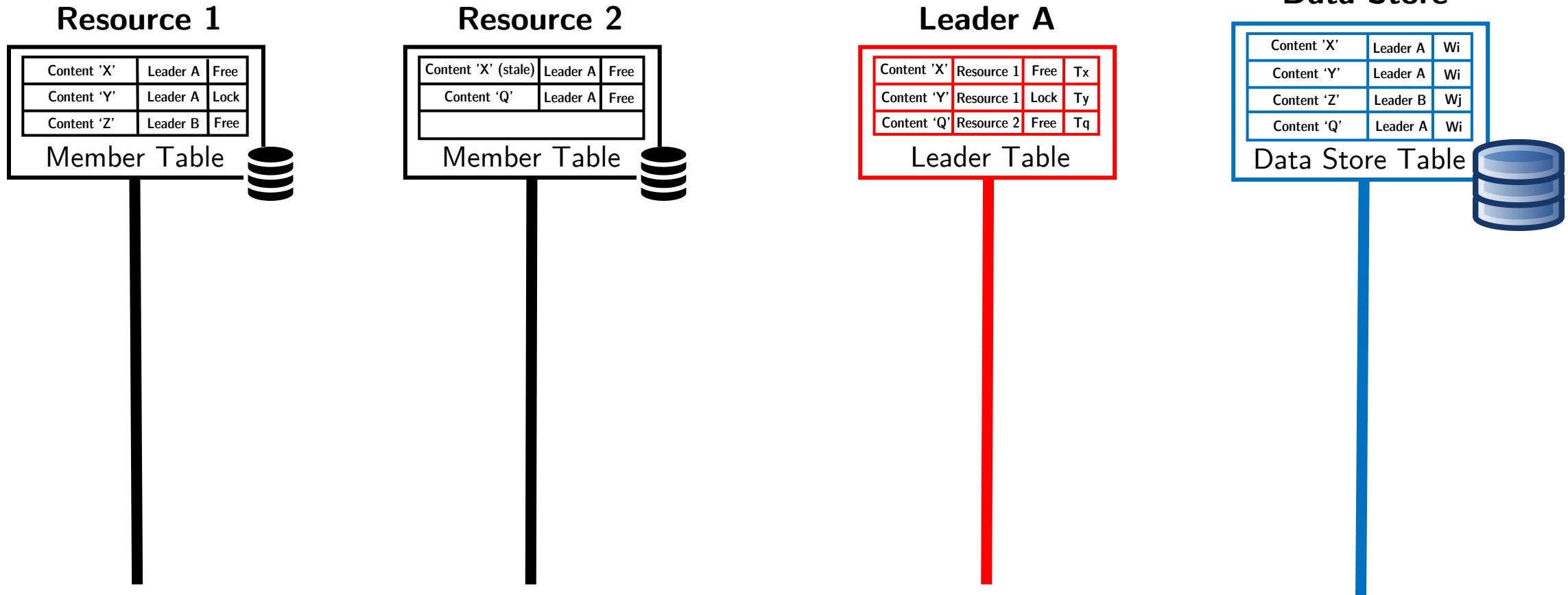
3. Data Store table

Maps data to groups

Data Name	Leader Address	Classifier Type
-----------	----------------	-----------------

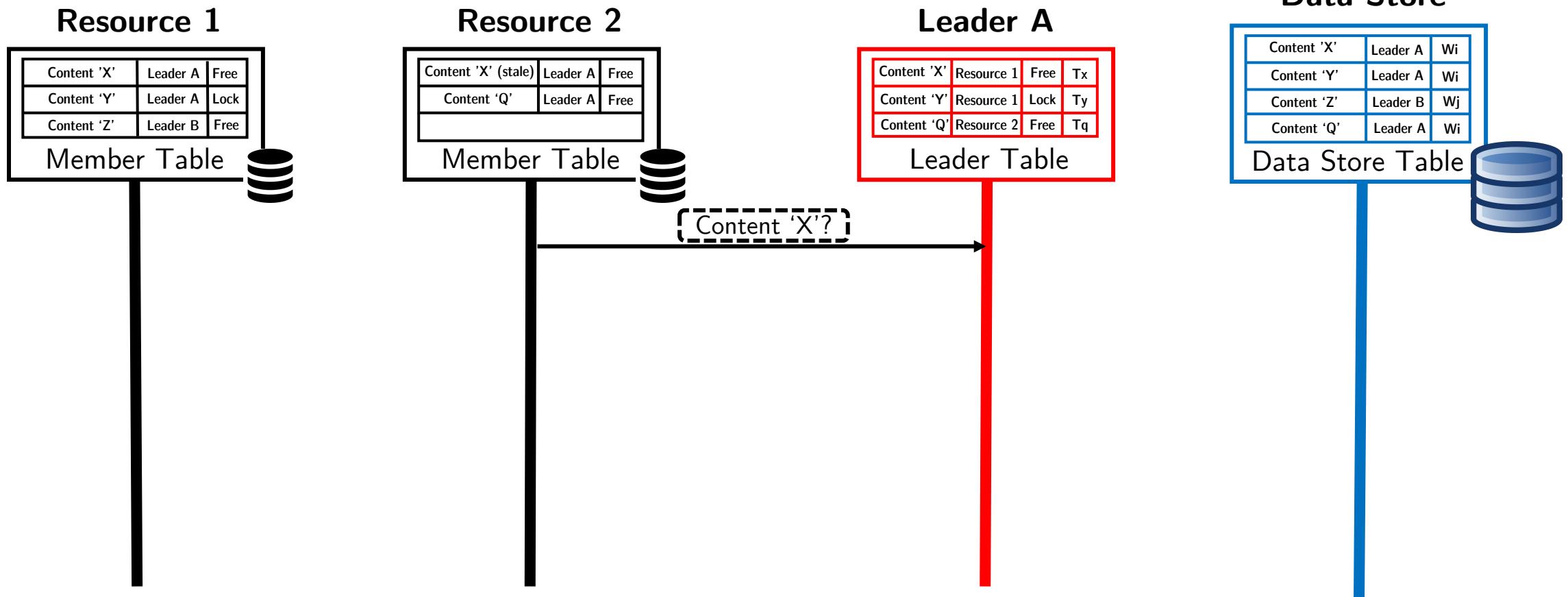
Content Retrieval in Group

Scenario 1: Retrieving free data



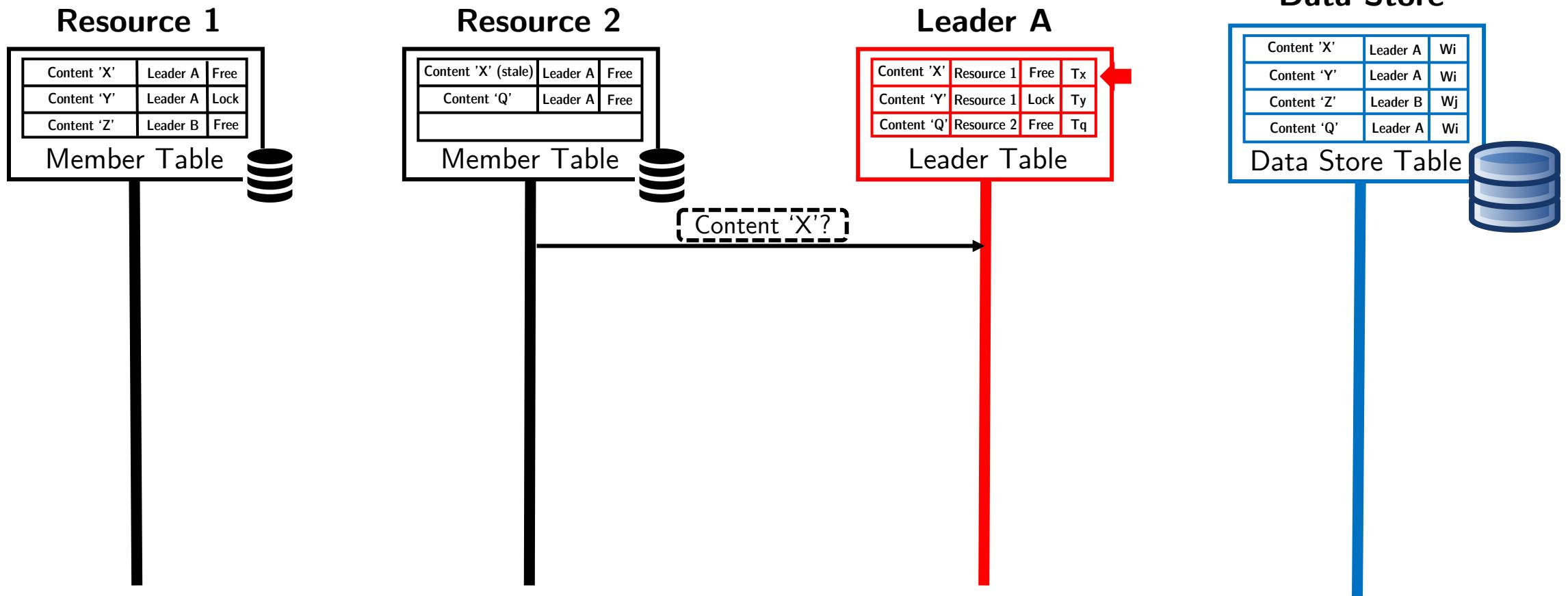
Content Retrieval in Group

Scenario 1: Retrieving free data



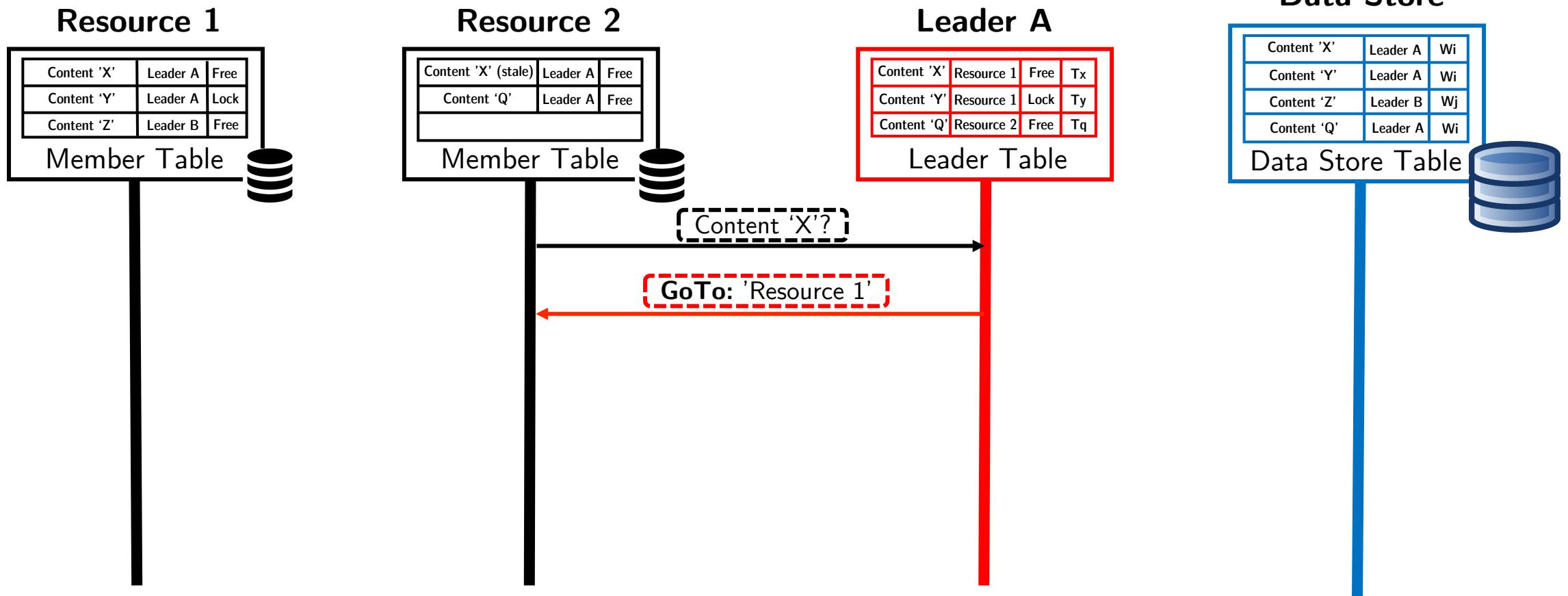
Content Retrieval in Group

Scenario 1: Retrieving free data



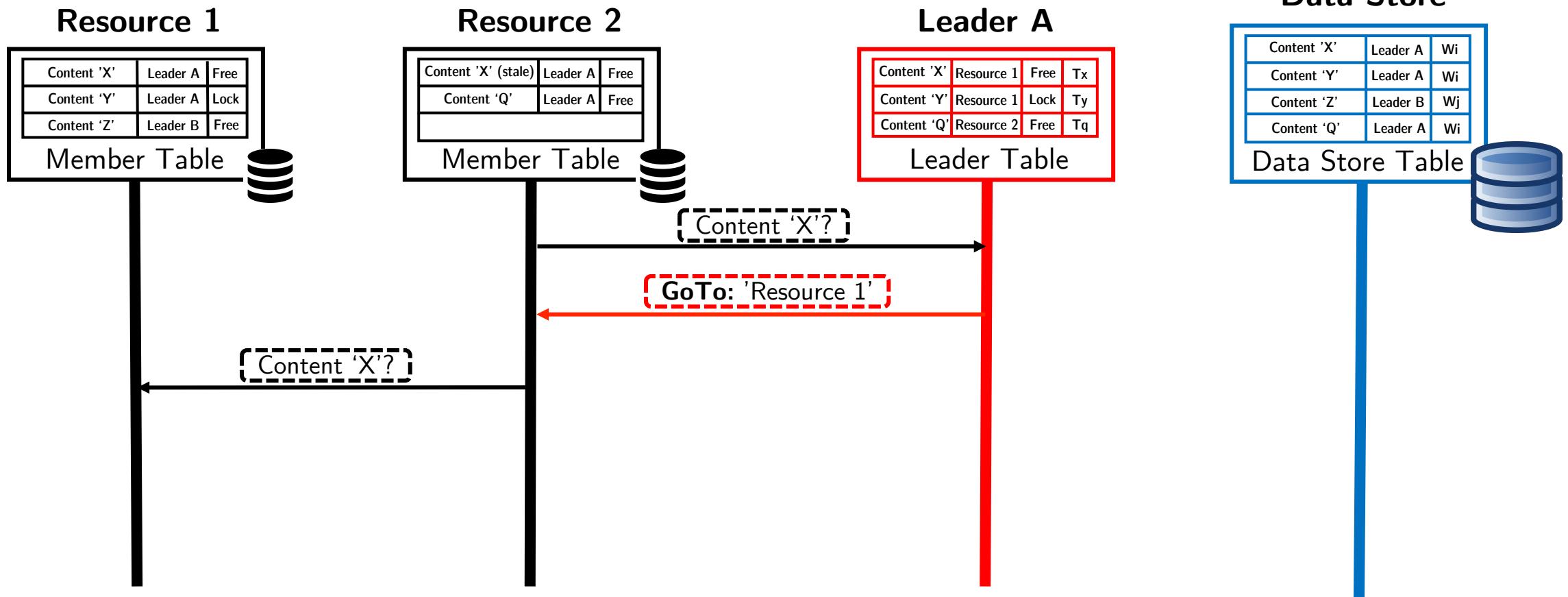
Content Retrieval in Group

Scenario 1: Retrieving free data



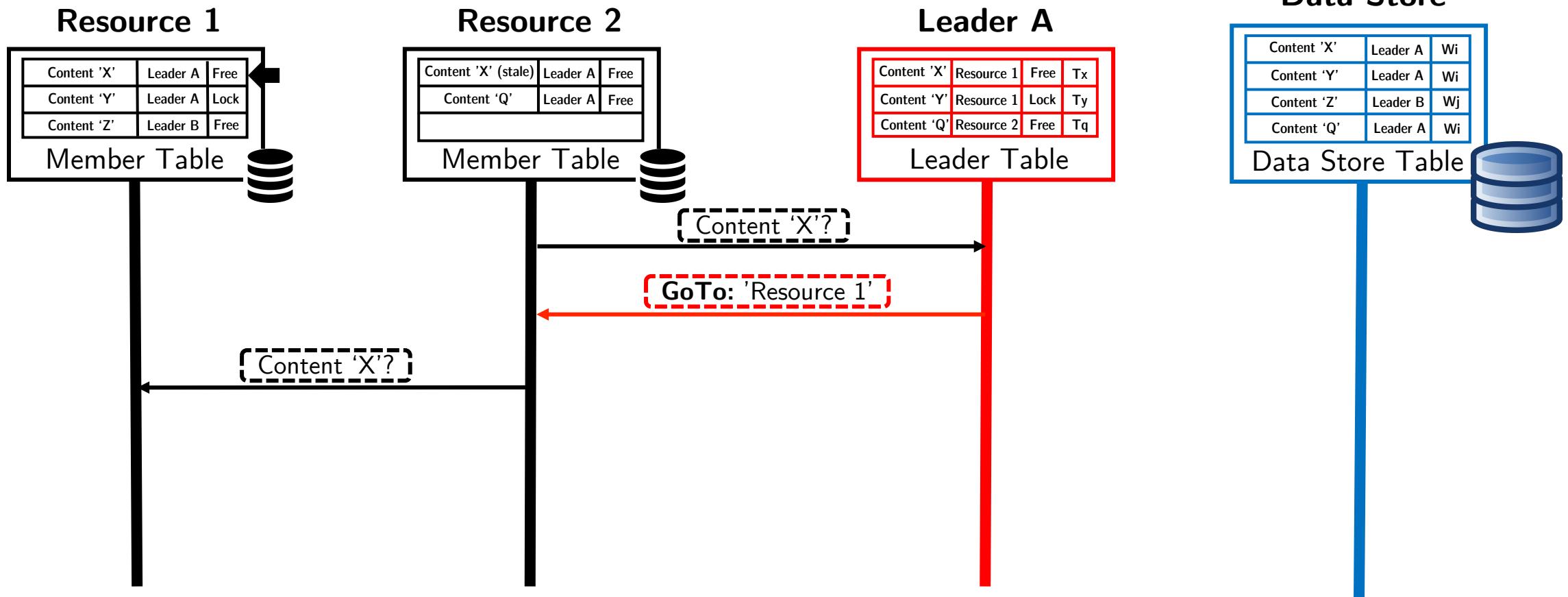
Content Retrieval in Group

Scenario 1: Retrieving free data



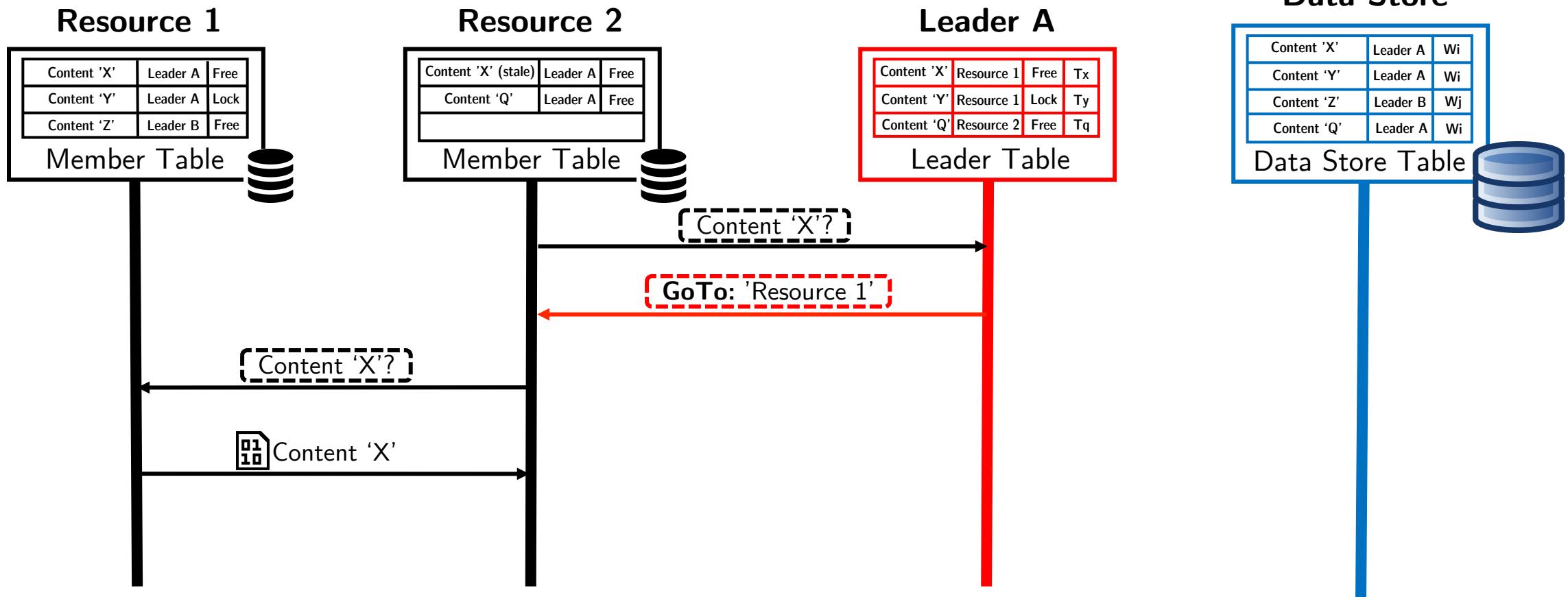
Content Retrieval in Group

Scenario 1: Retrieving free data



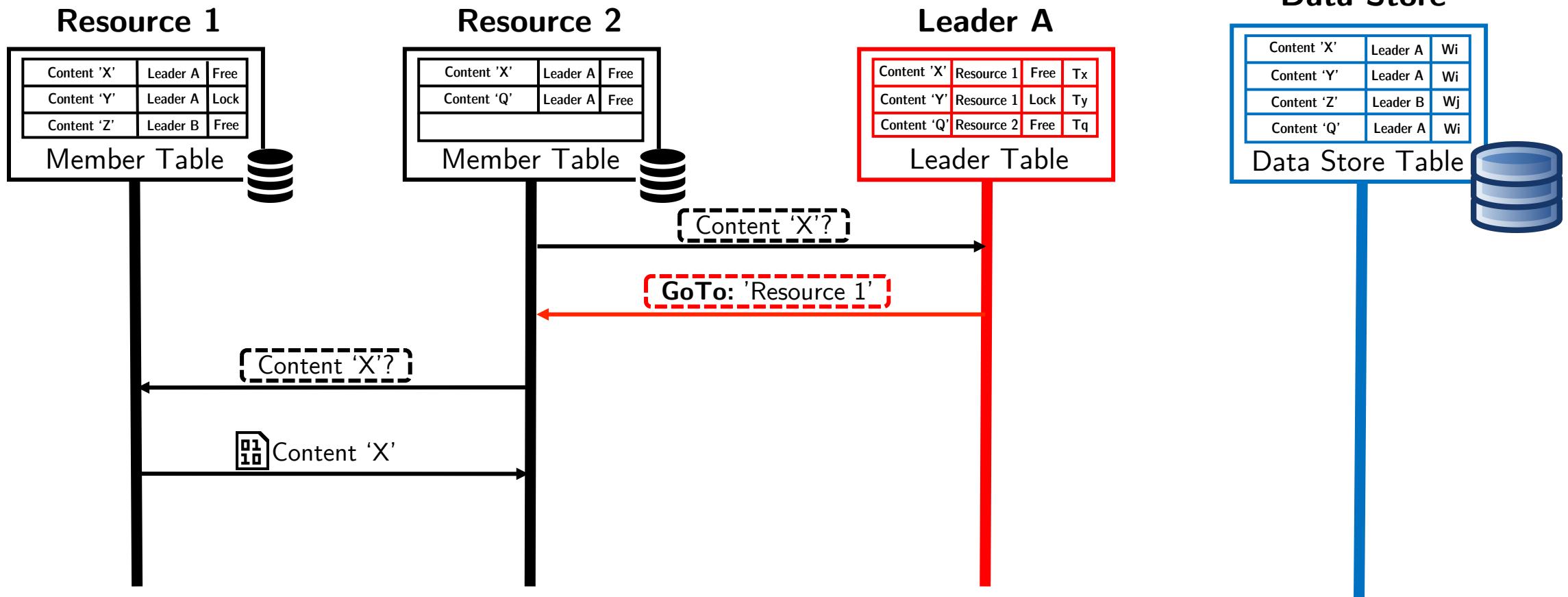
Content Retrieval in Group

Scenario 1: Retrieving free data



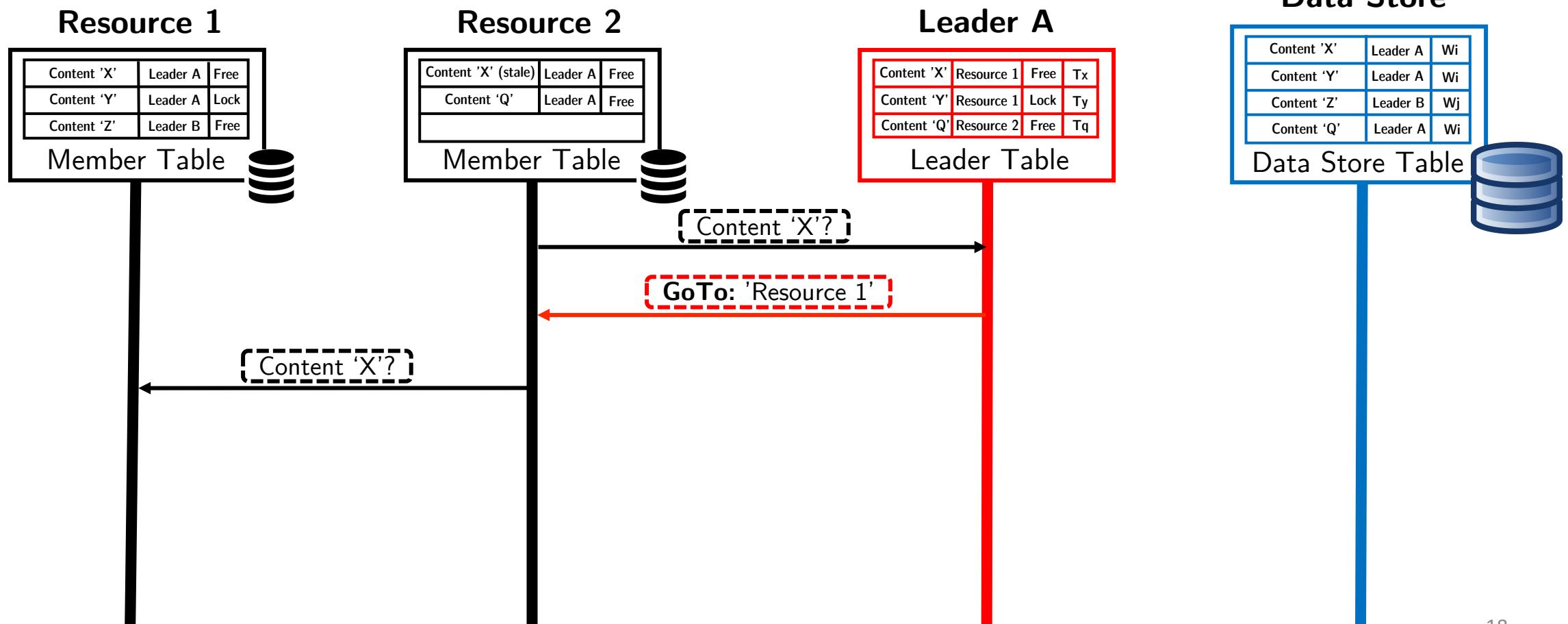
Content Retrieval in Group

Scenario 1: Retrieving free data



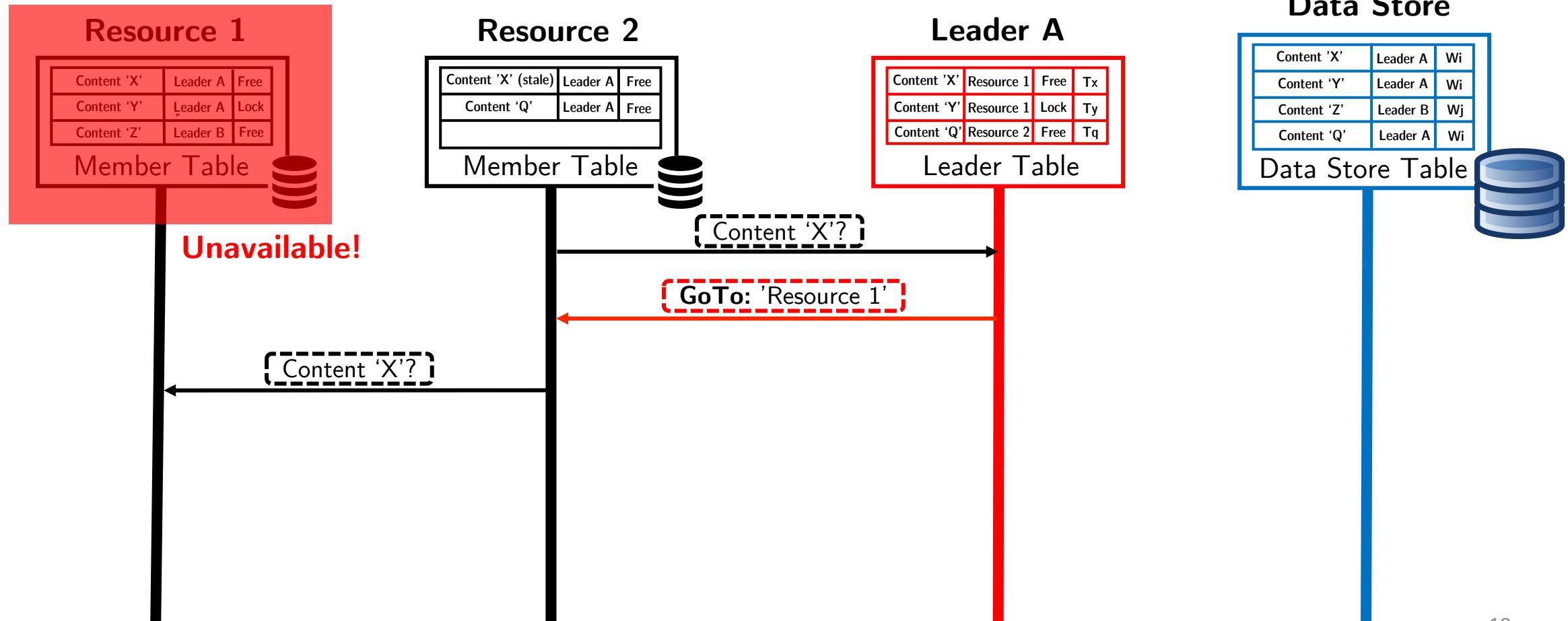
Content Retrieval in Group

Scenario 2: Resource failure



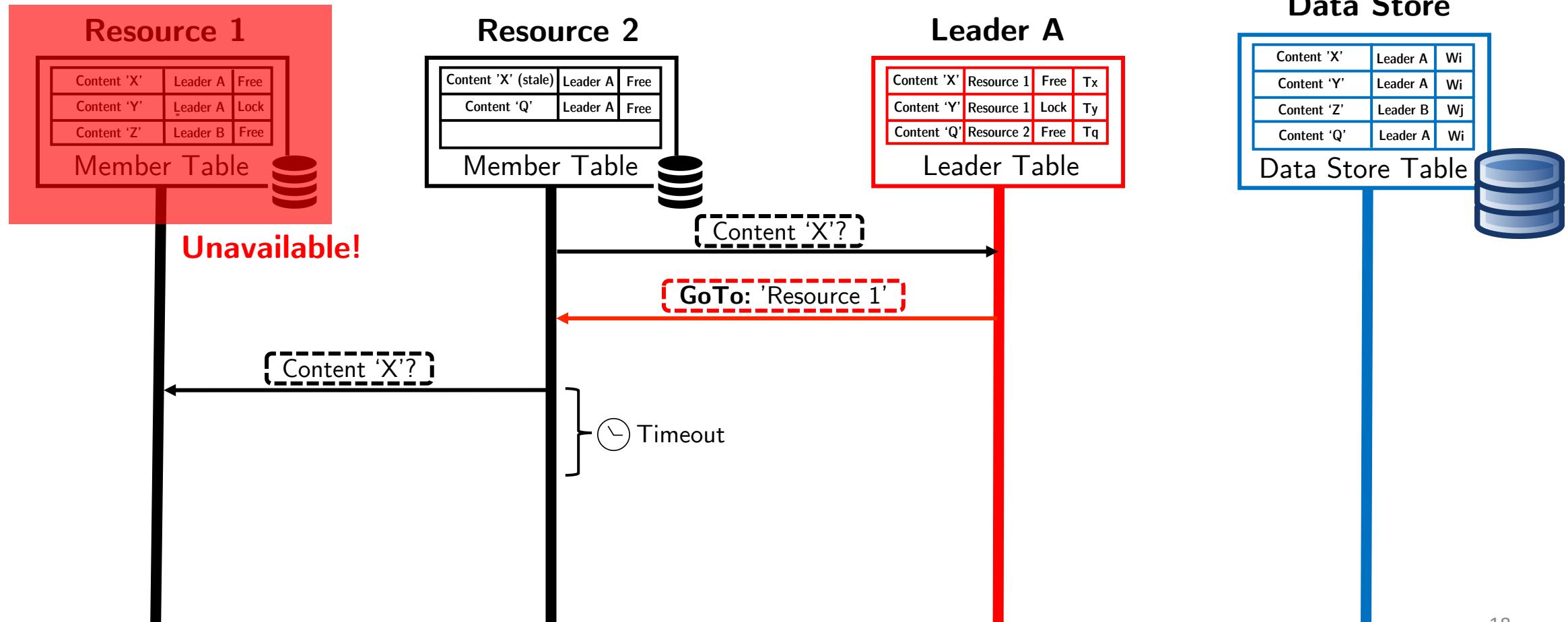
Content Retrieval in Group

Scenario 2: Resource failure



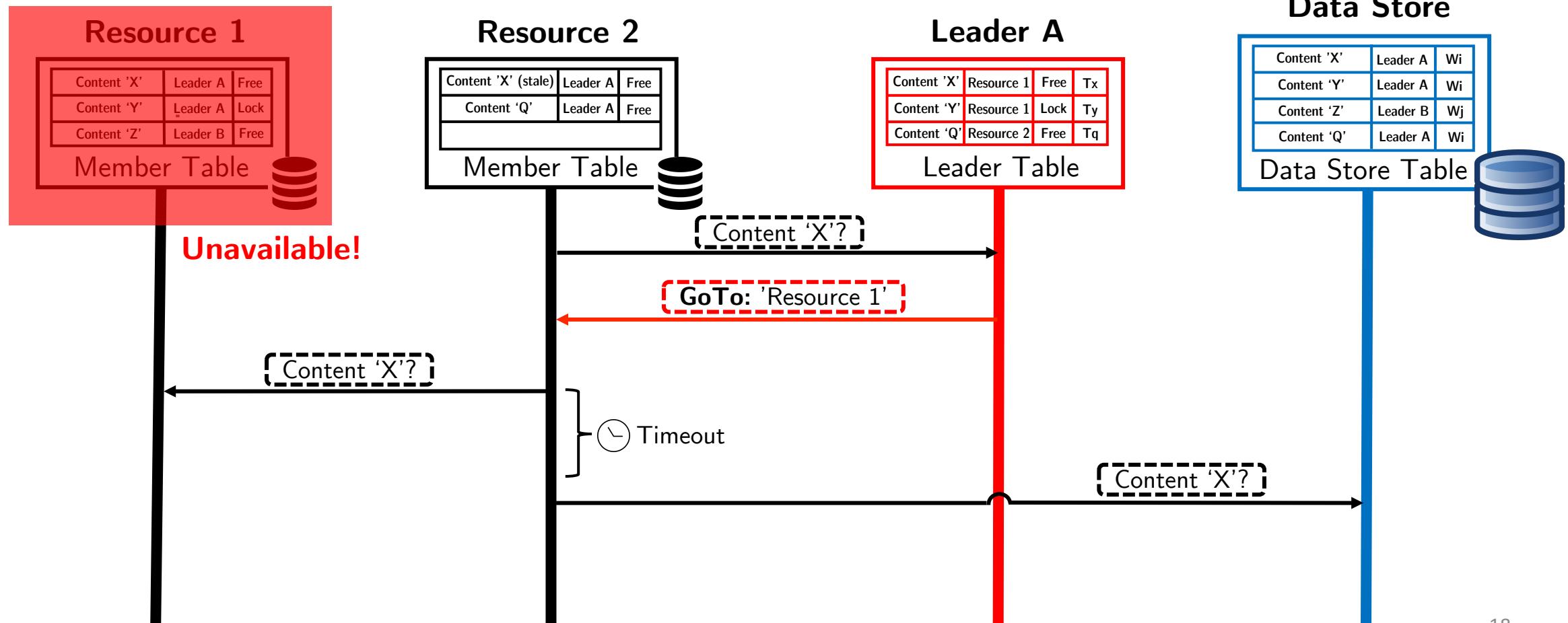
Content Retrieval in Group

Scenario 2: Resource failure



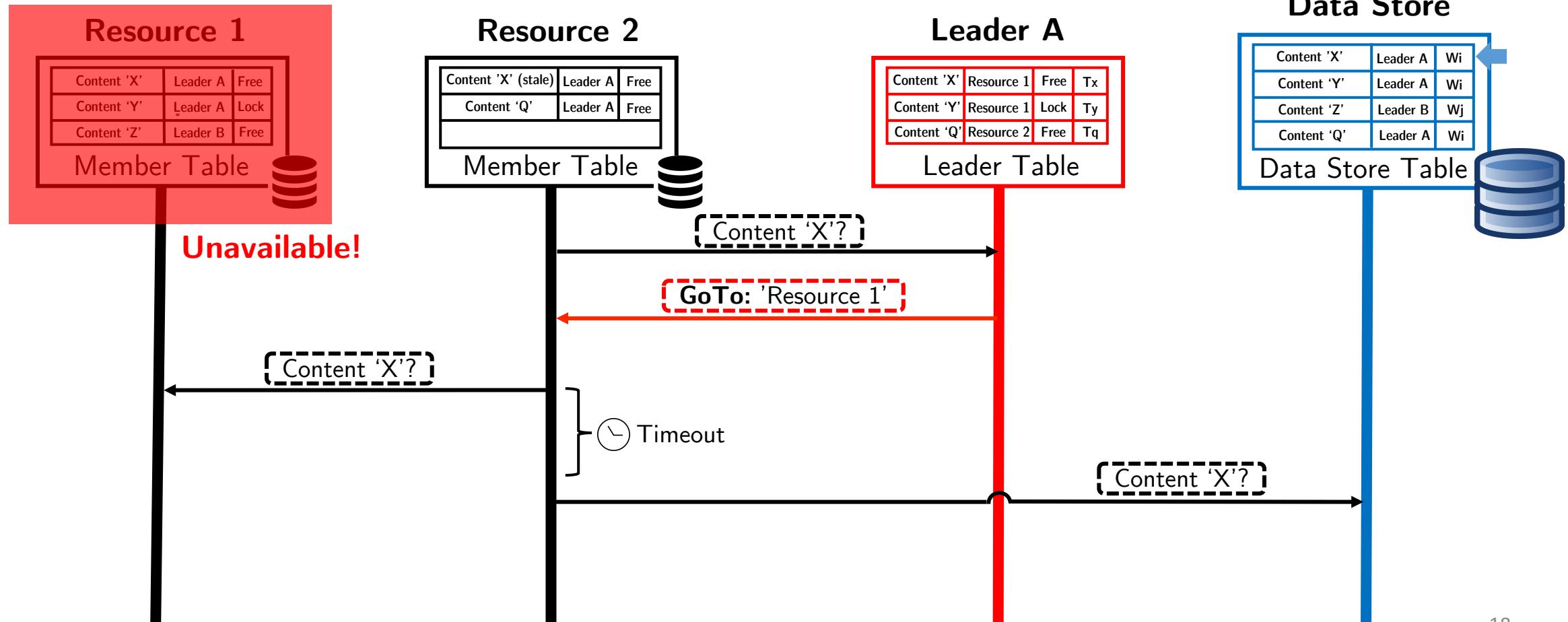
Content Retrieval in Group

Scenario 2: Resource failure



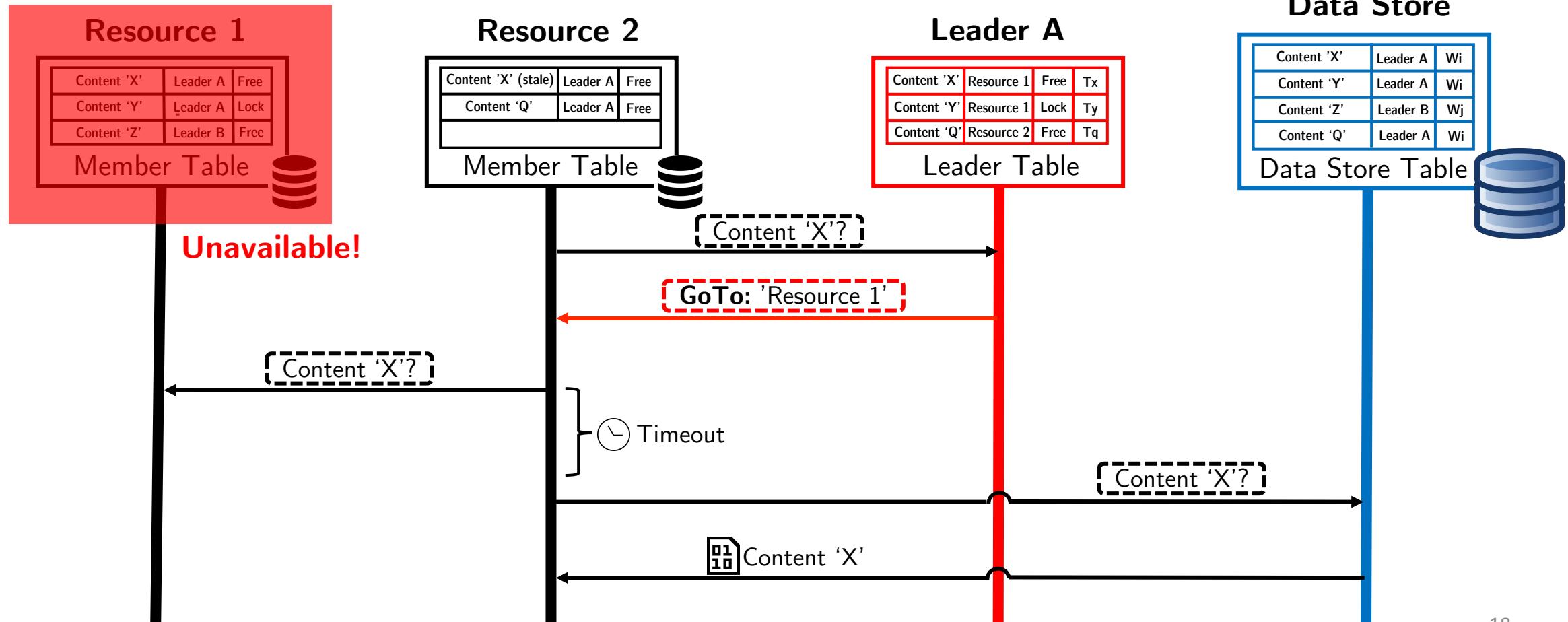
Content Retrieval in Group

Scenario 2: Resource failure



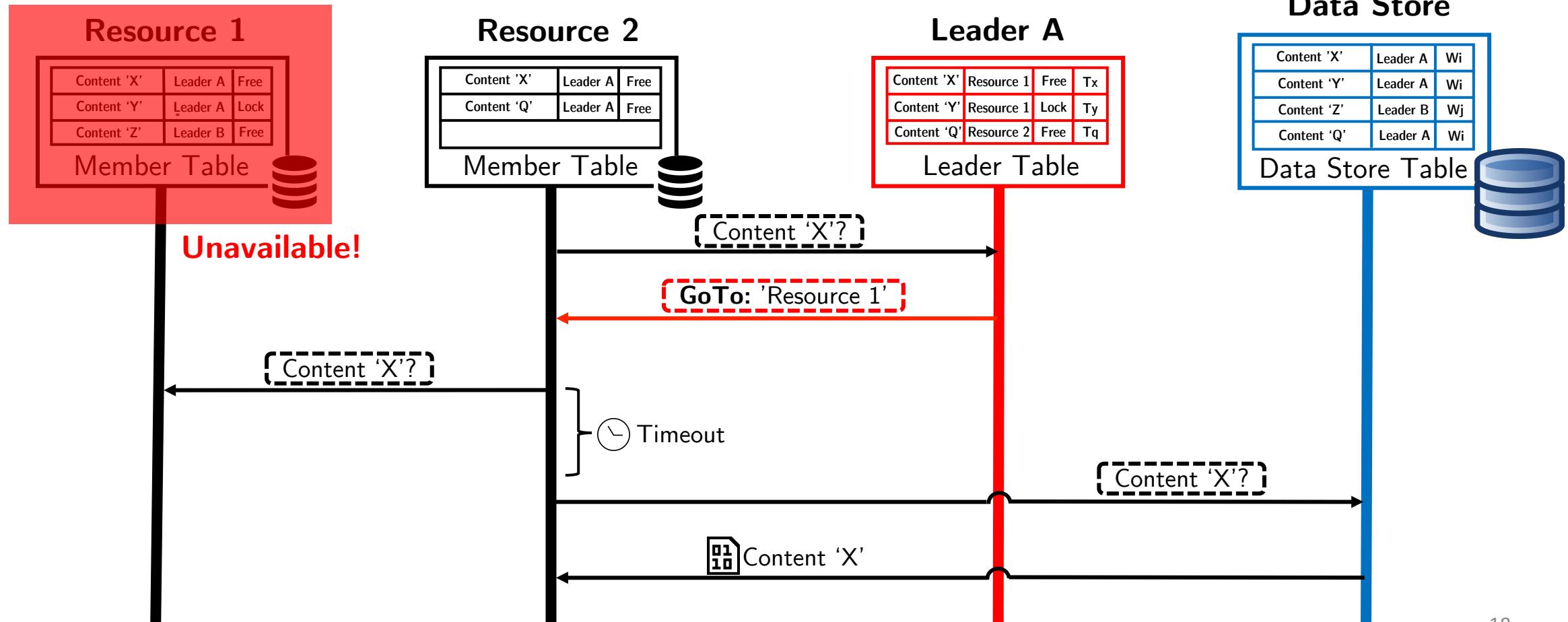
Content Retrieval in Group

Scenario 2: Resource failure



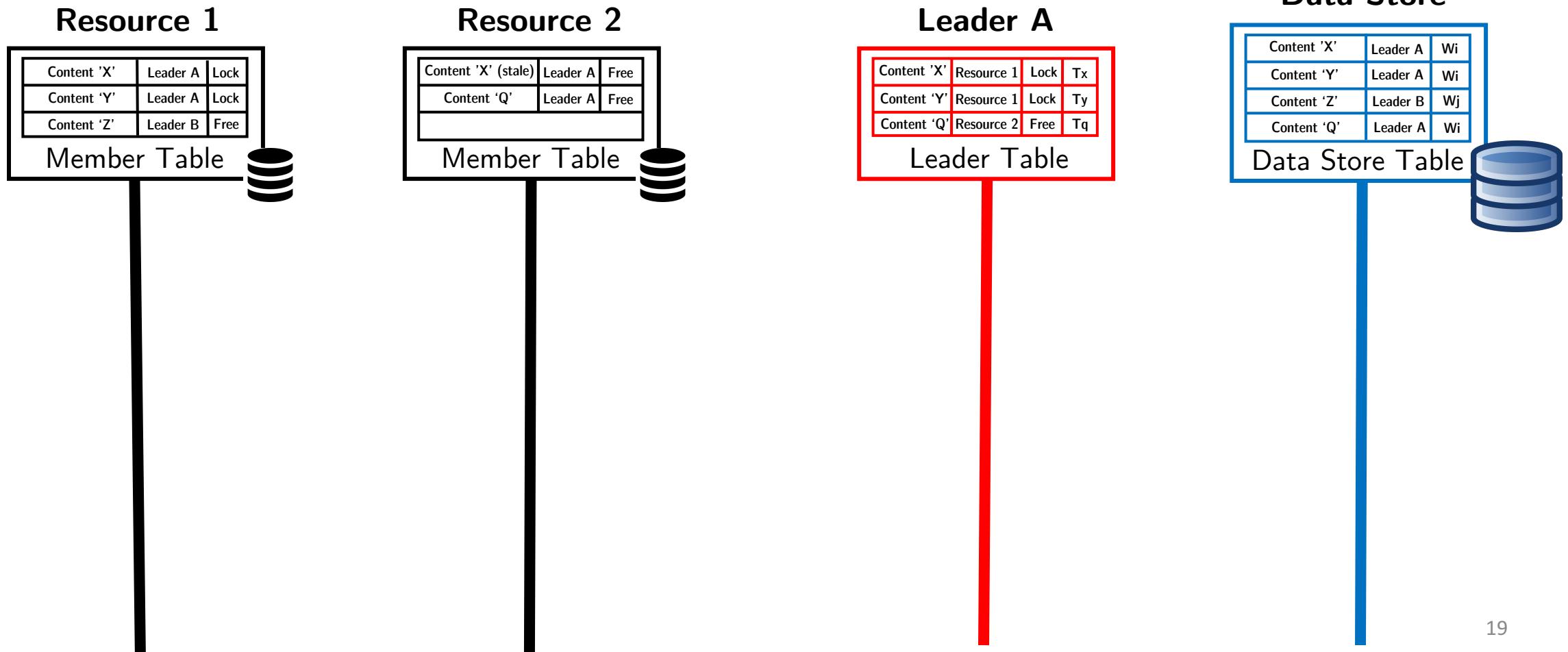
Content Retrieval in Group

Scenario 2: Resource failure



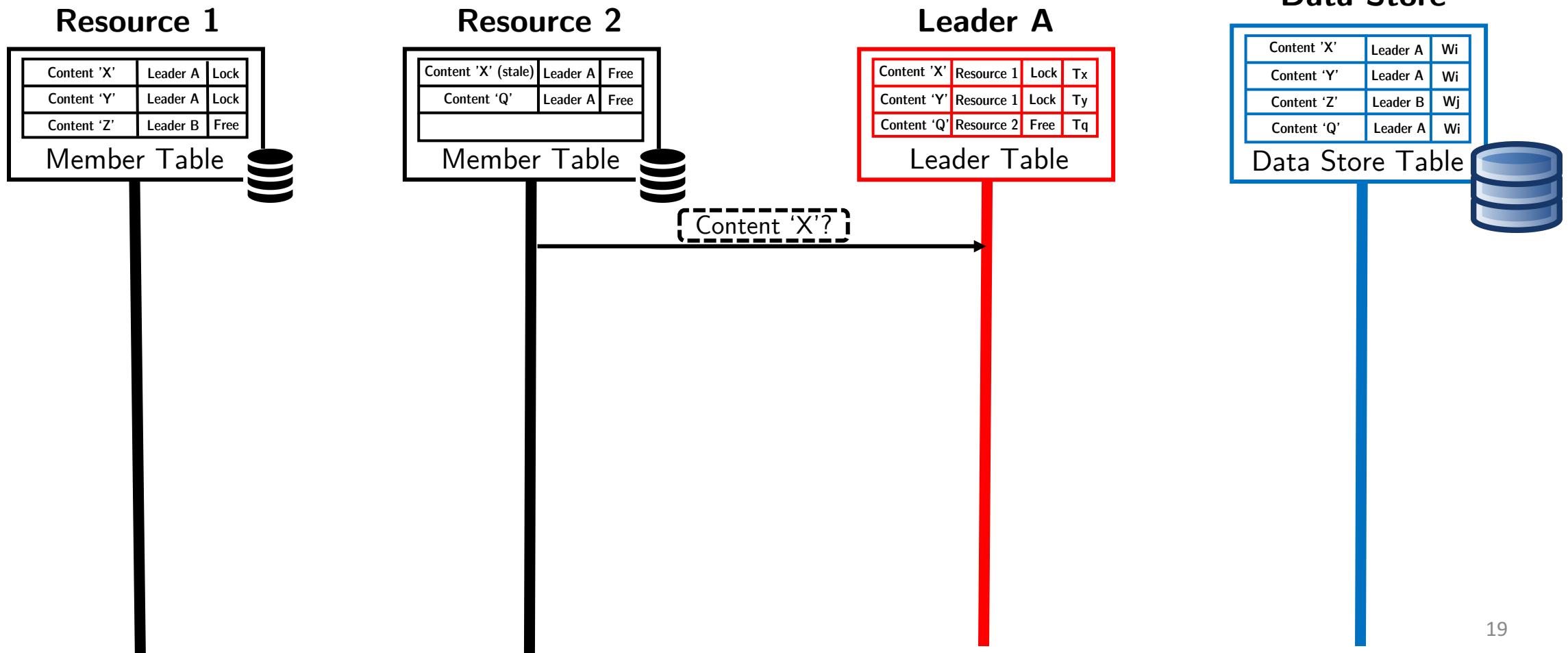
Content Retrieval in Group

Scenario 3: Data under update



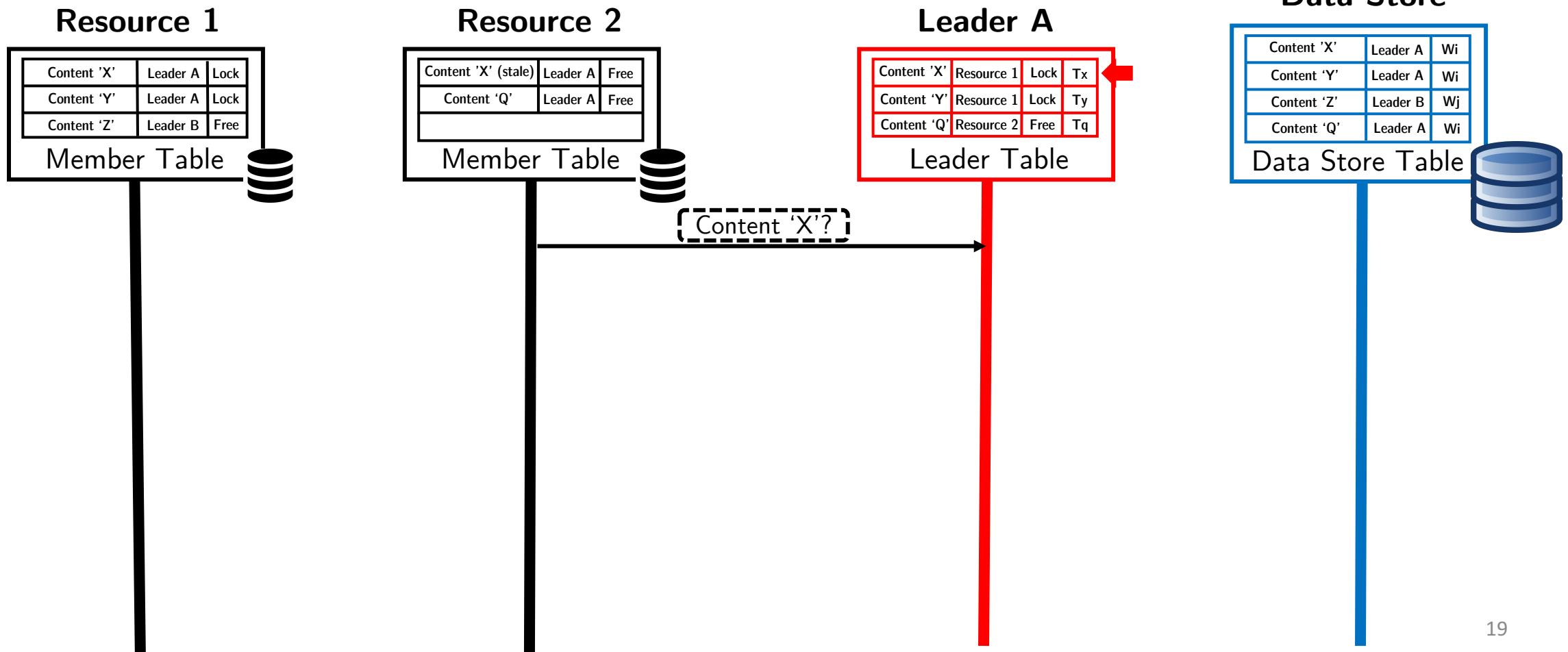
Content Retrieval in Group

Scenario 3: Data under update



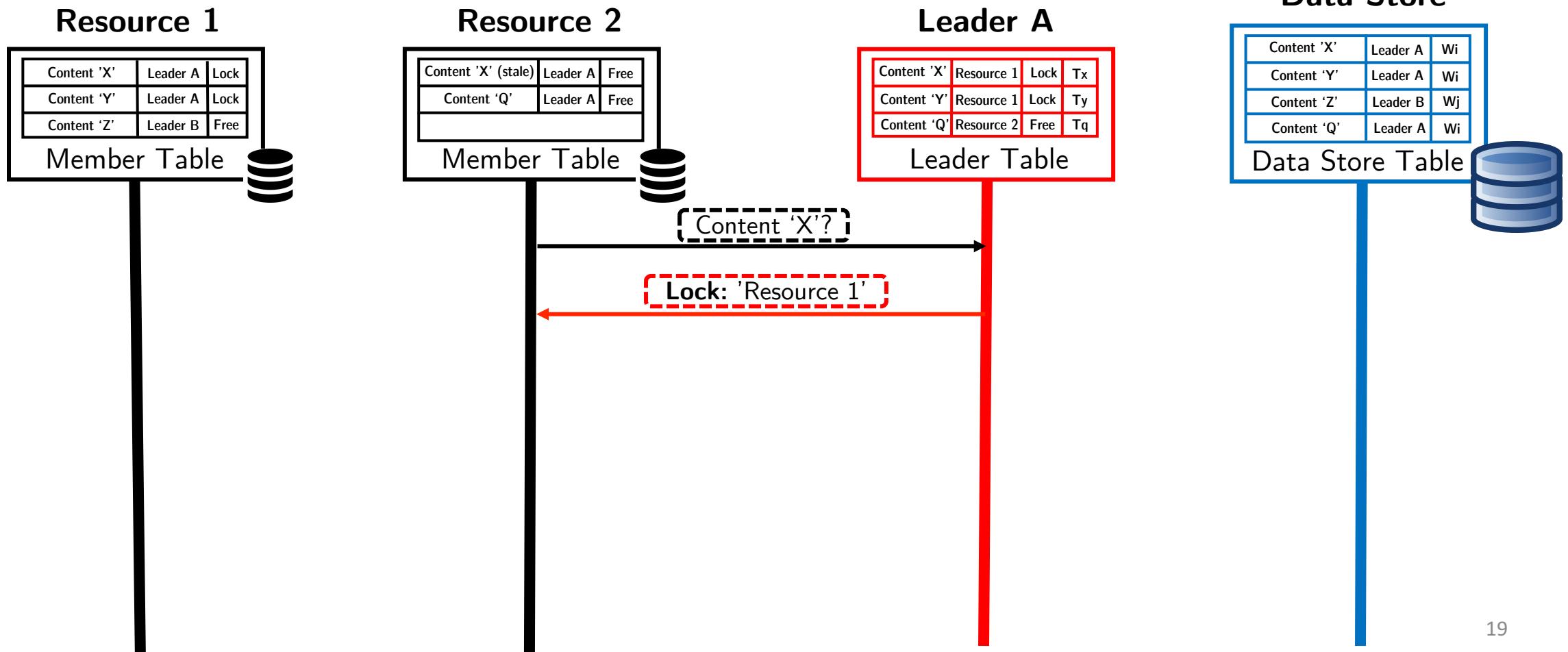
Content Retrieval in Group

Scenario 3: Data under update



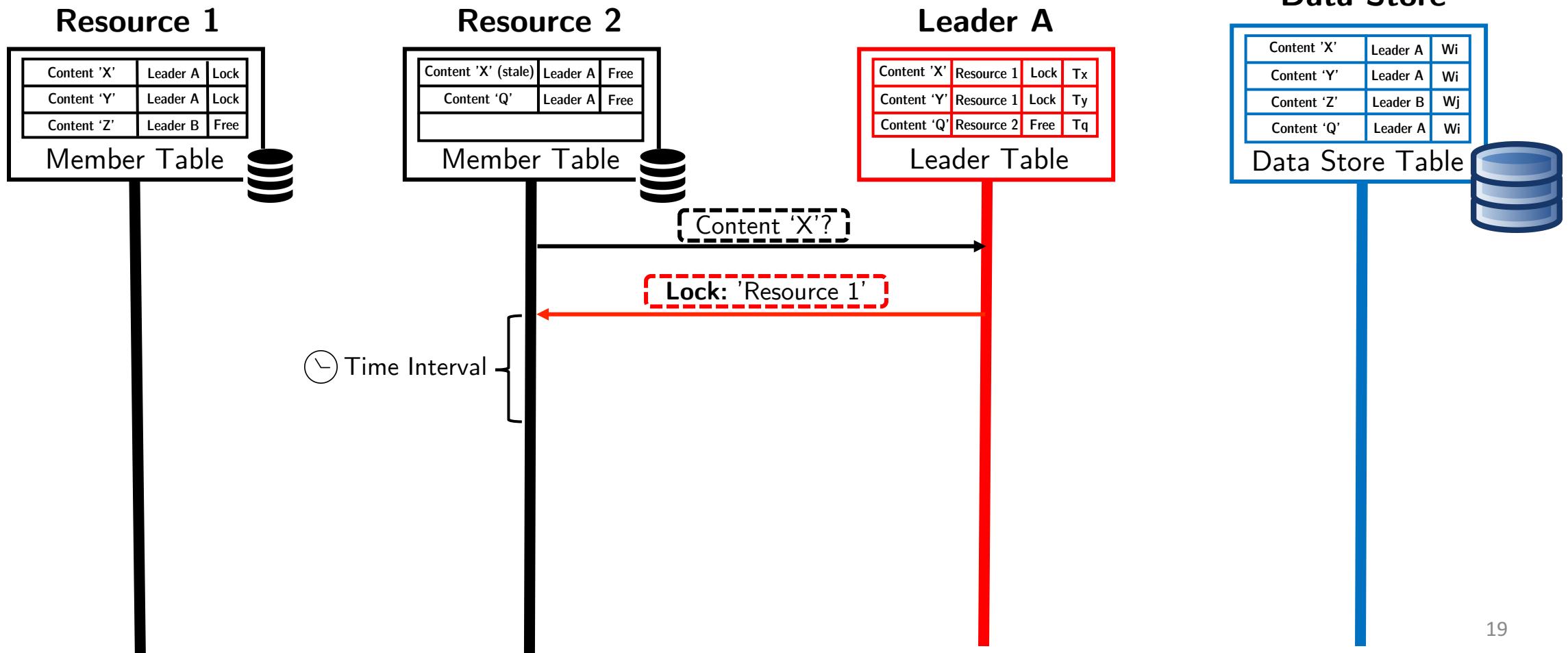
Content Retrieval in Group

Scenario 3: Data under update



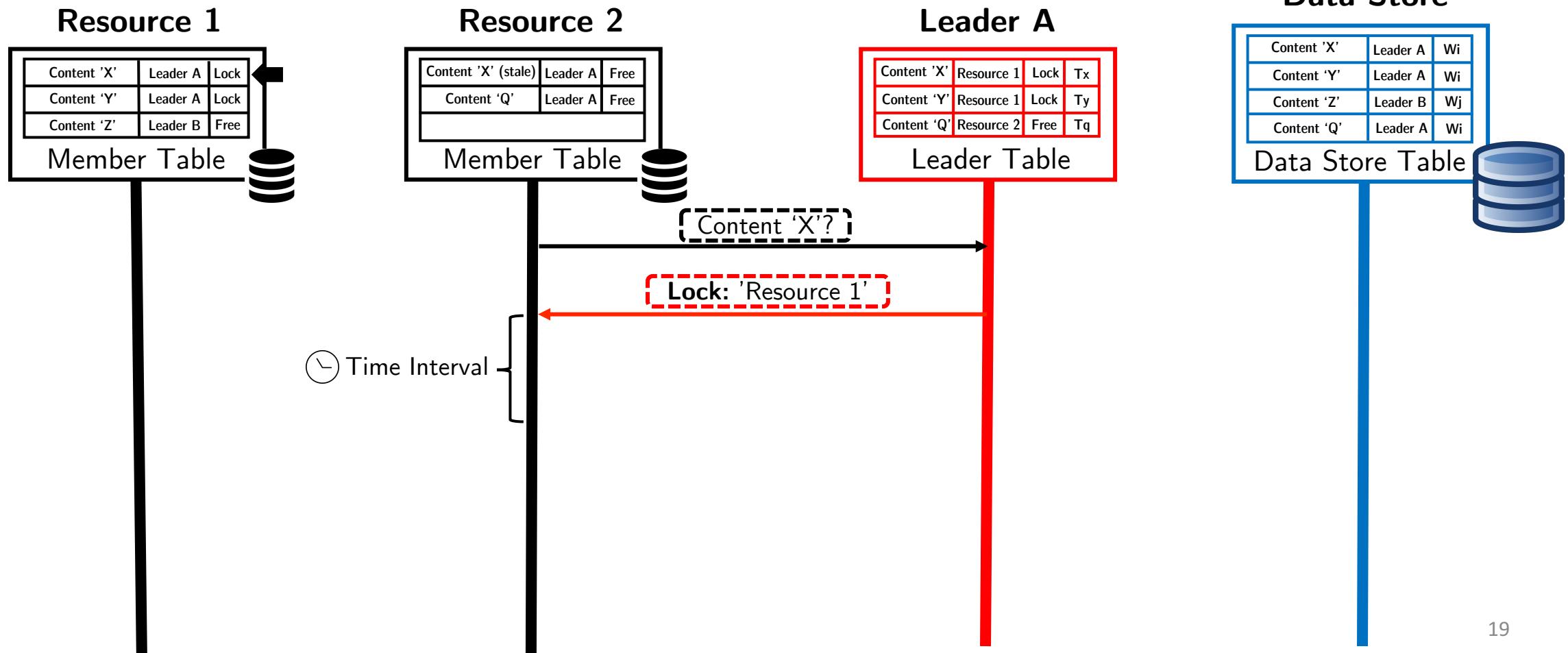
Content Retrieval in Group

Scenario 3: Data under update



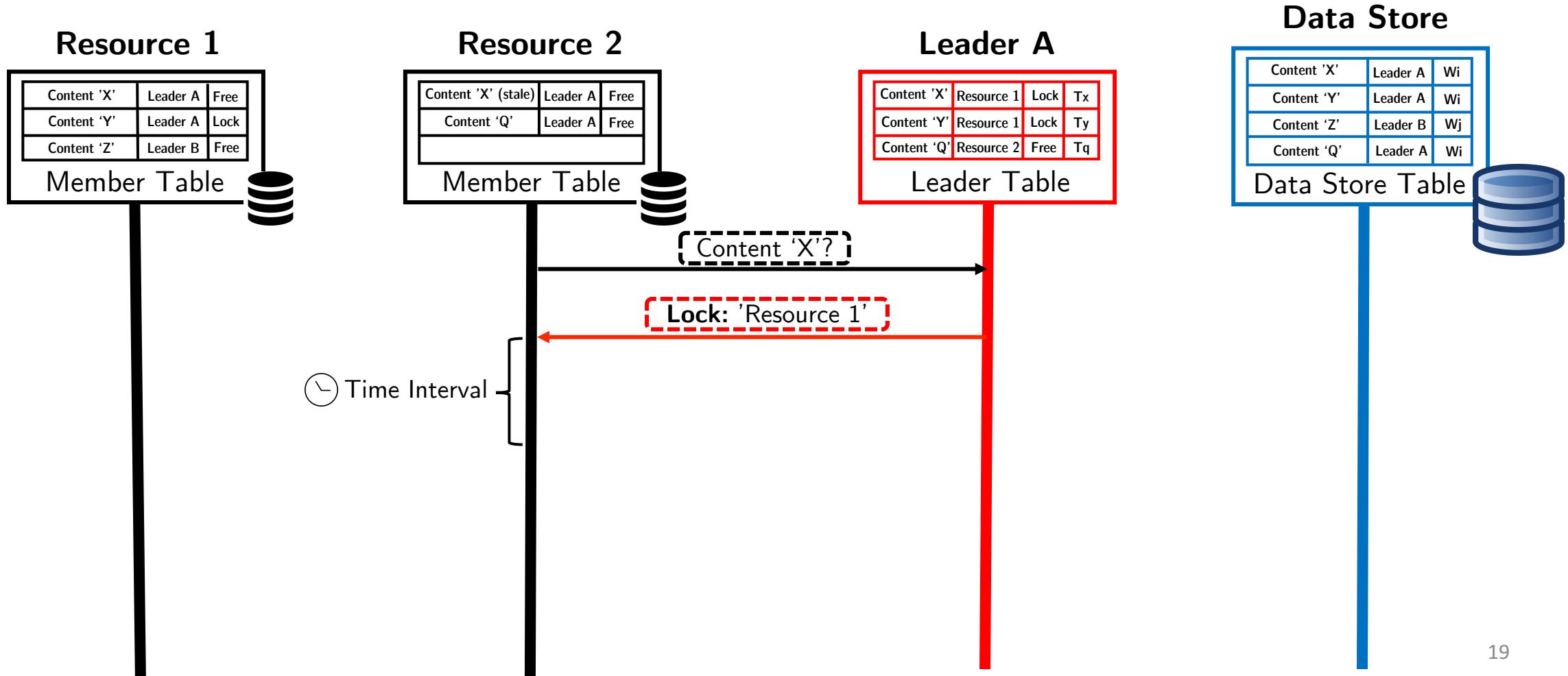
Content Retrieval in Group

Scenario 3: Data under update



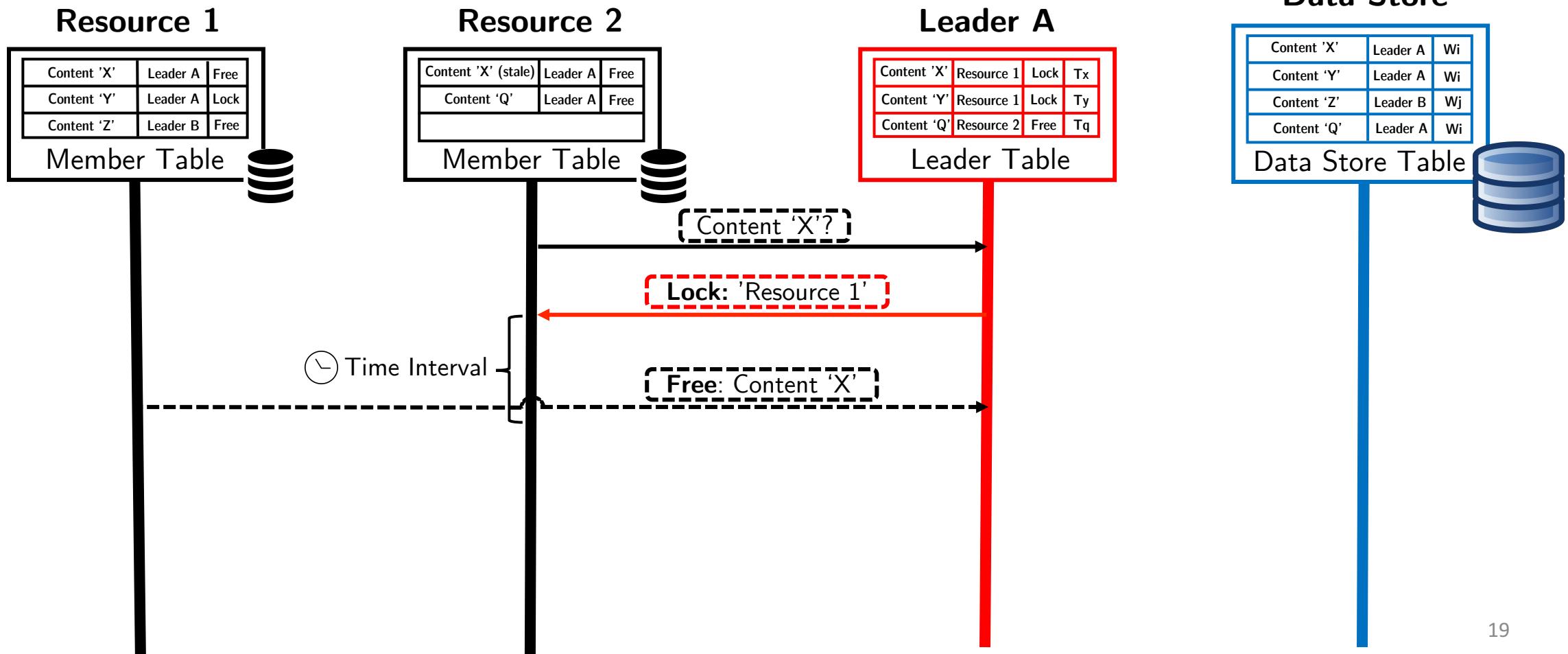
Content Retrieval in Group

Scenario 3: Data under update



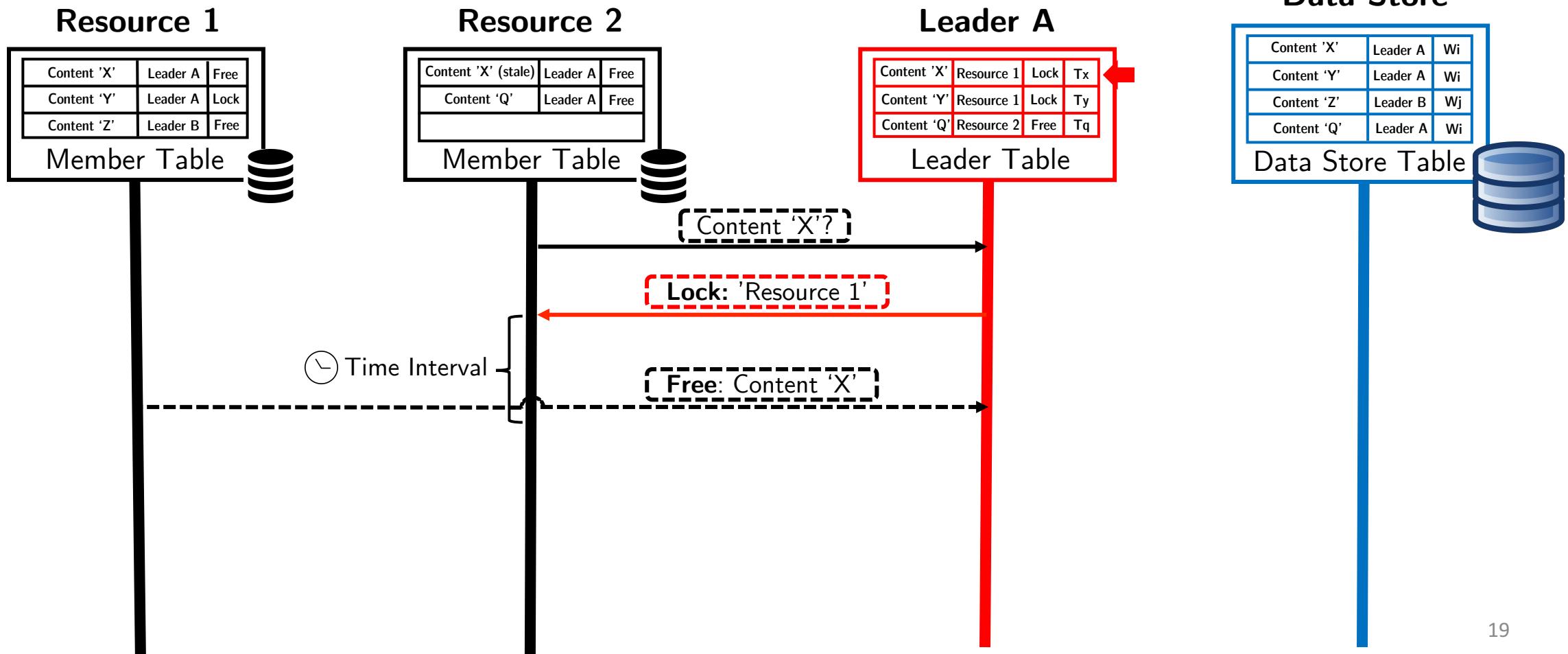
Content Retrieval in Group

Scenario 3: Data under update



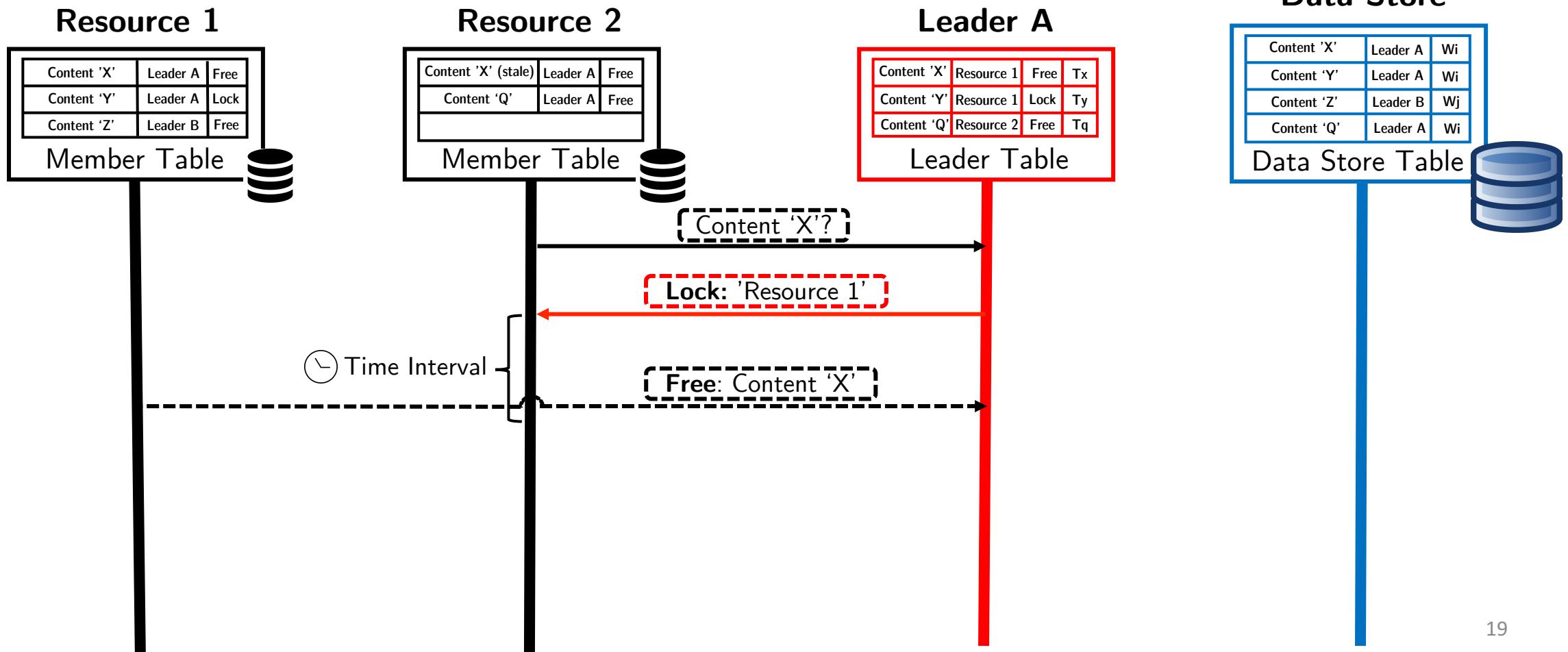
Content Retrieval in Group

Scenario 3: Data under update



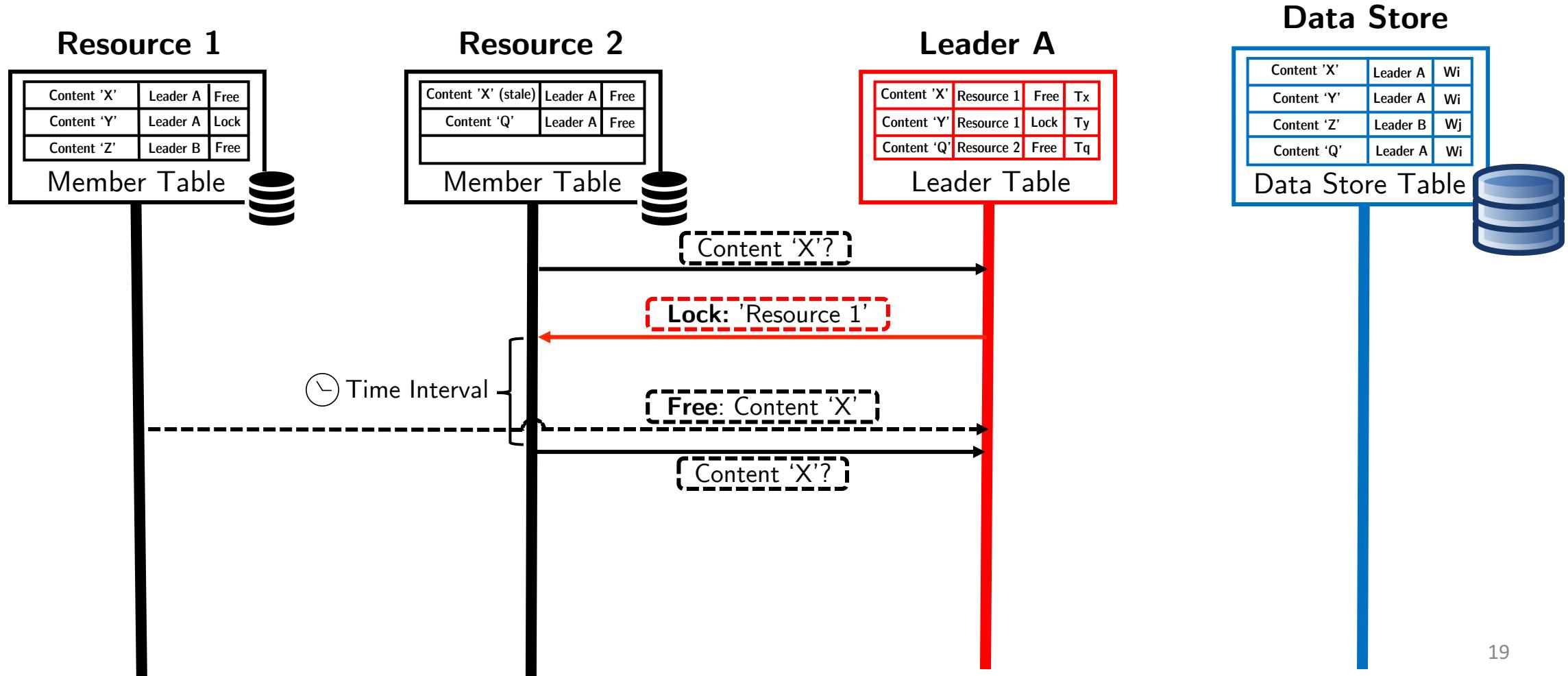
Content Retrieval in Group

Scenario 3: Data under update



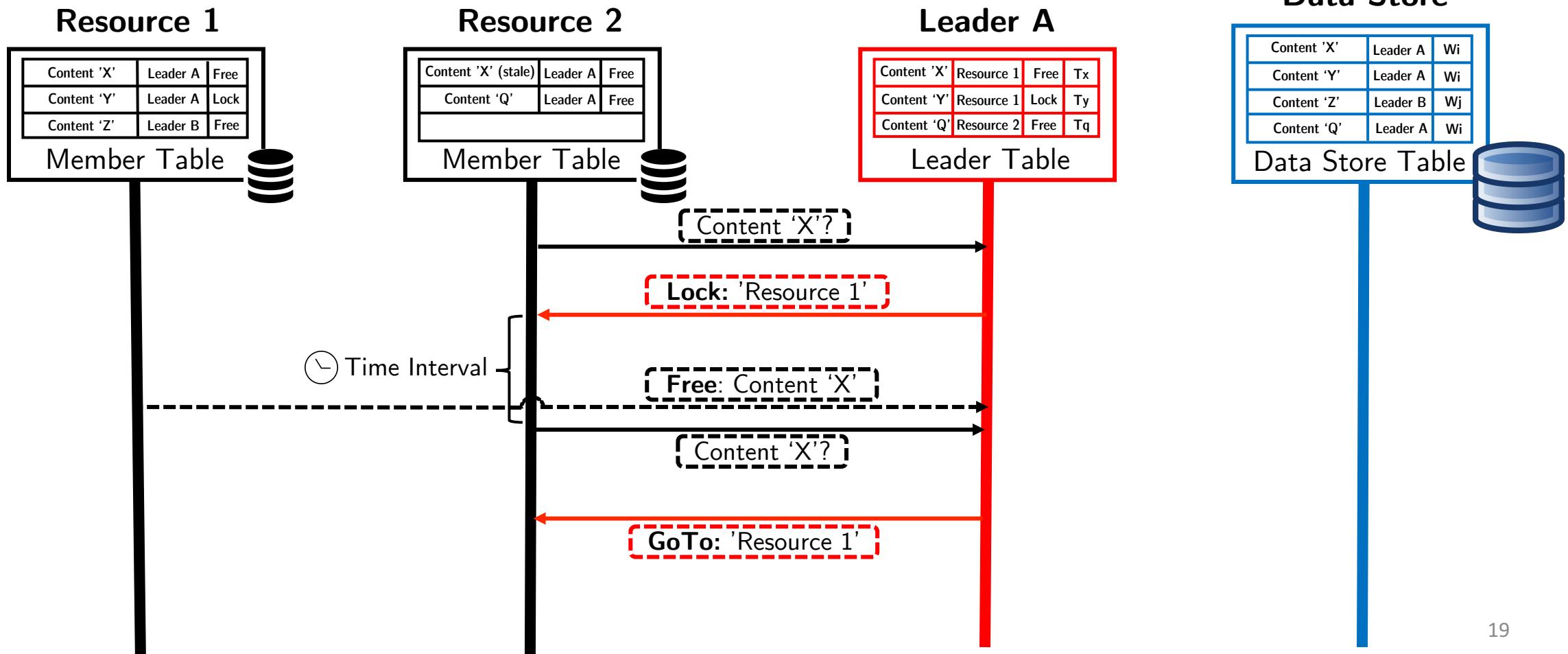
Content Retrieval in Group

Scenario 3: Data under update



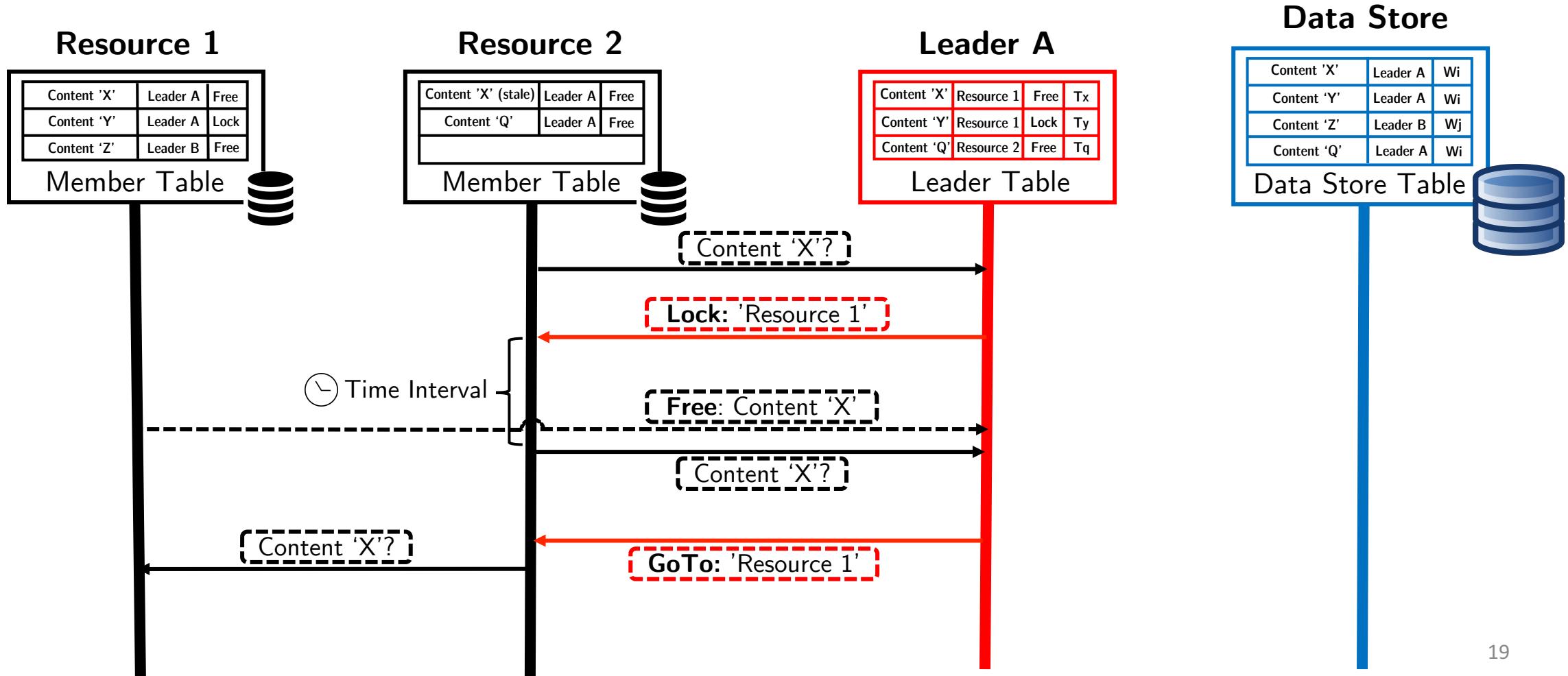
Content Retrieval in Group

Scenario 3: Data under update



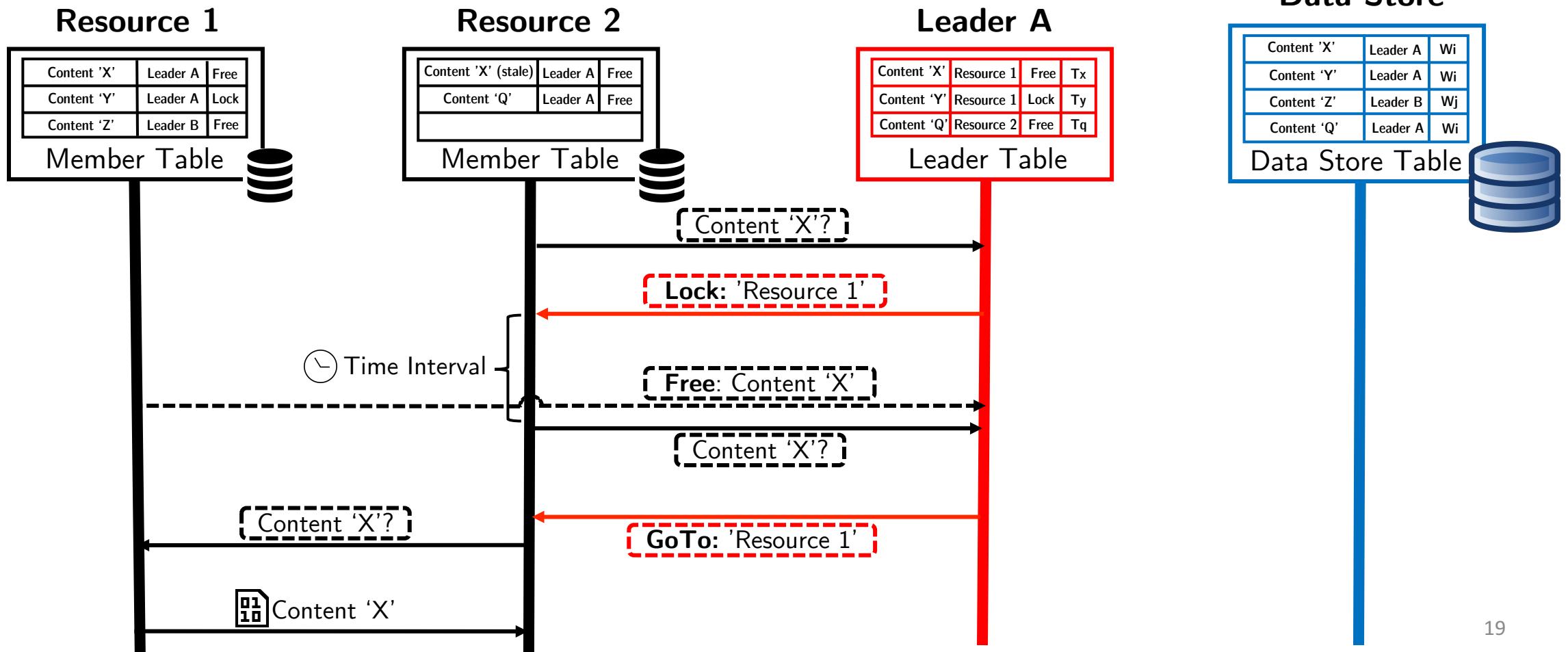
Content Retrieval in Group

Scenario 3: Data under update



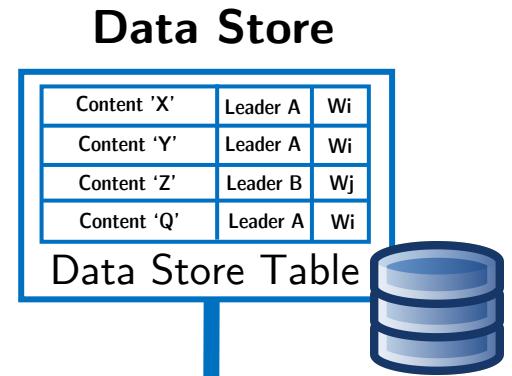
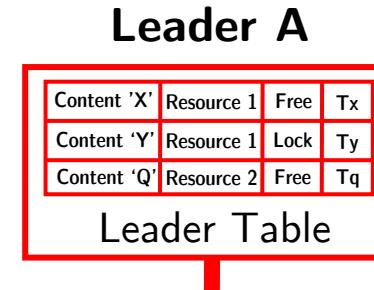
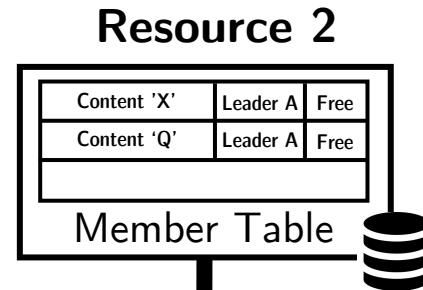
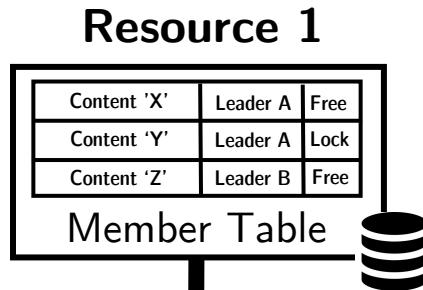
Content Retrieval in Group

Scenario 3: Data under update



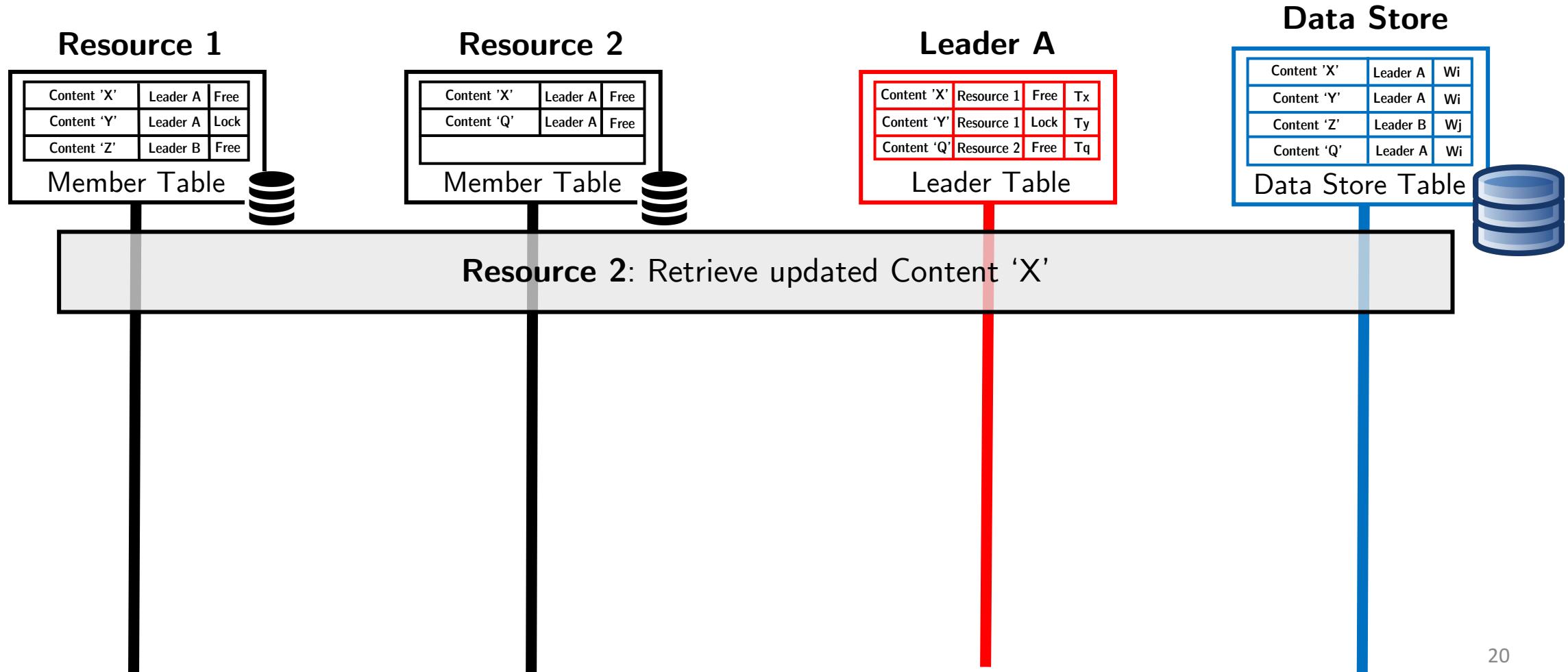
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



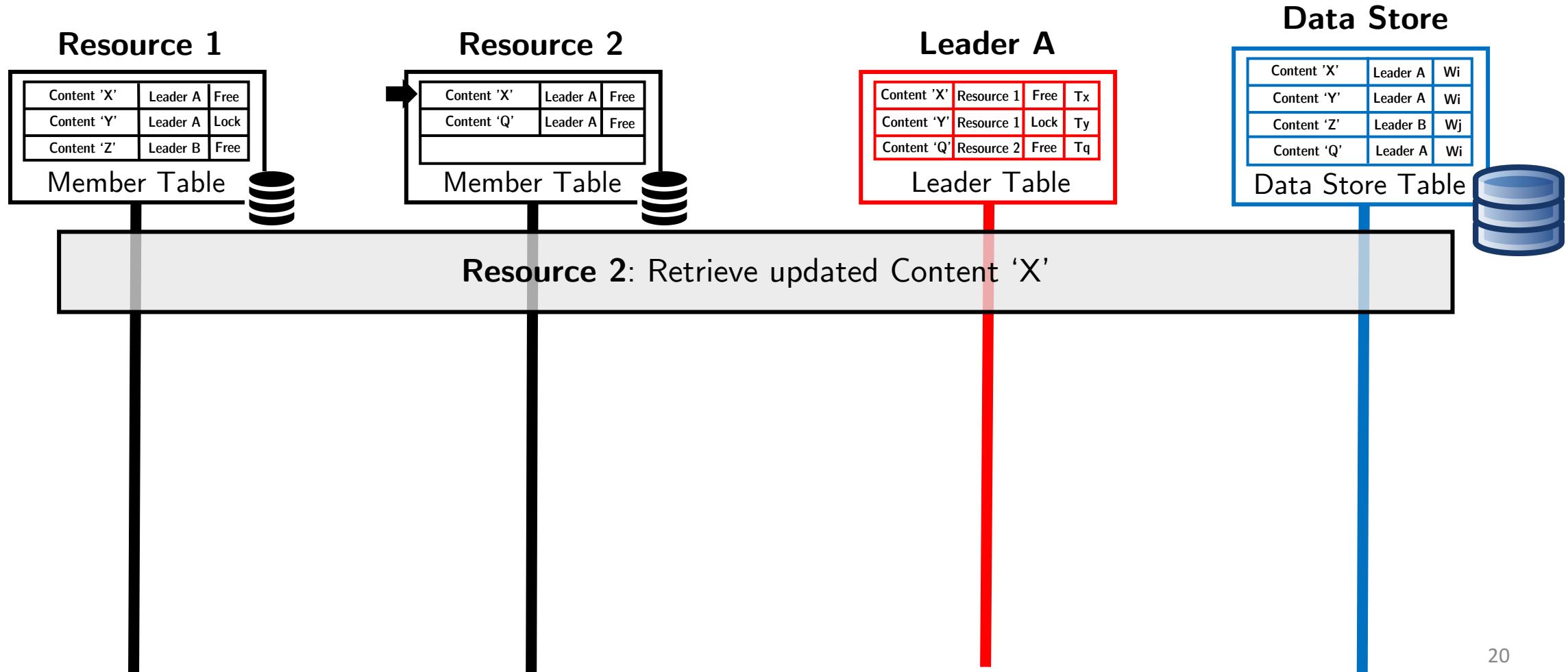
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



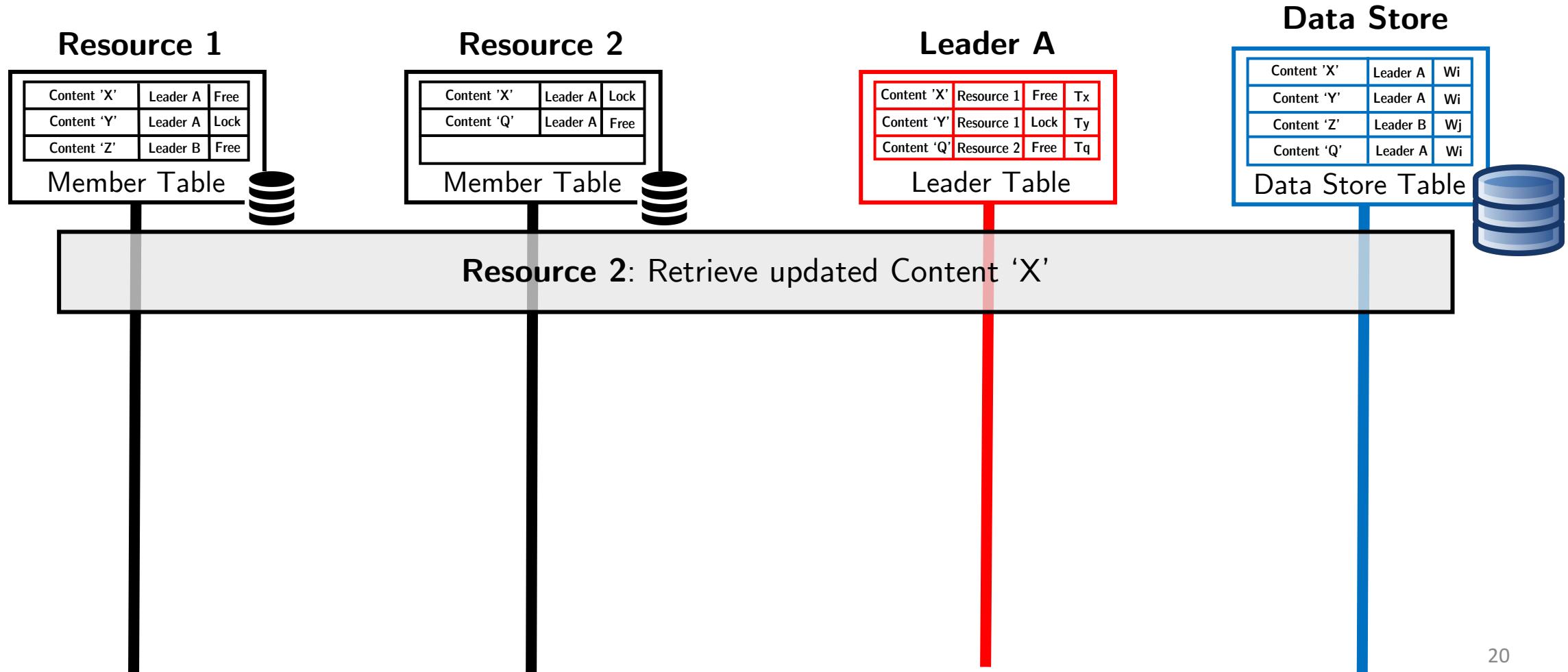
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



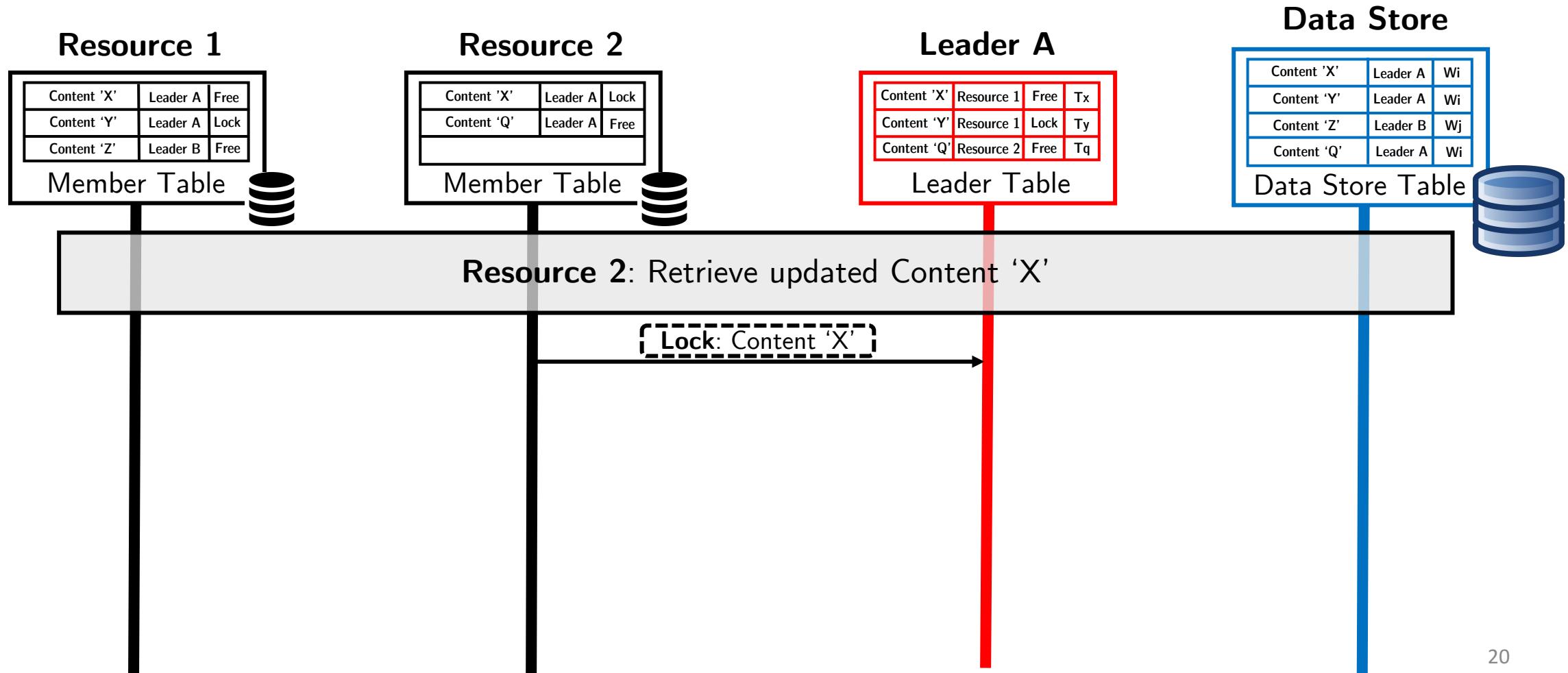
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



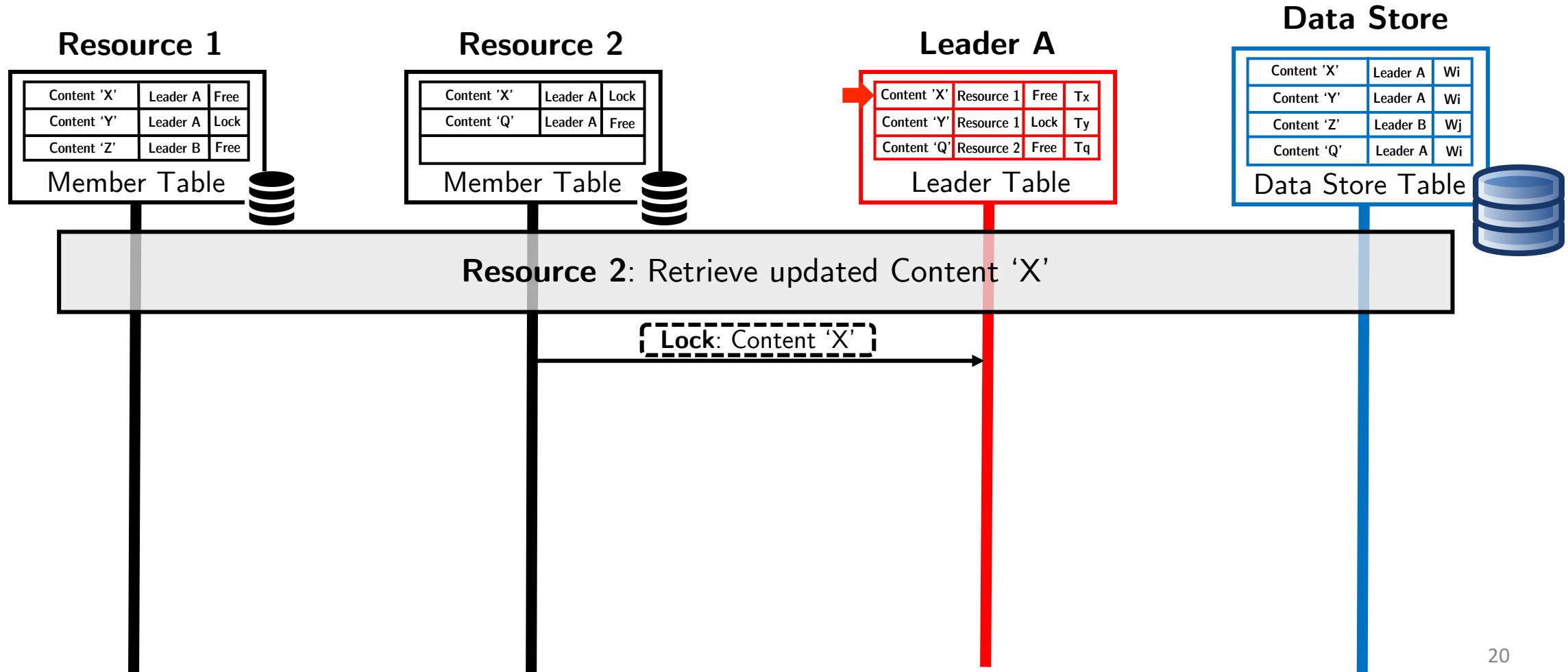
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



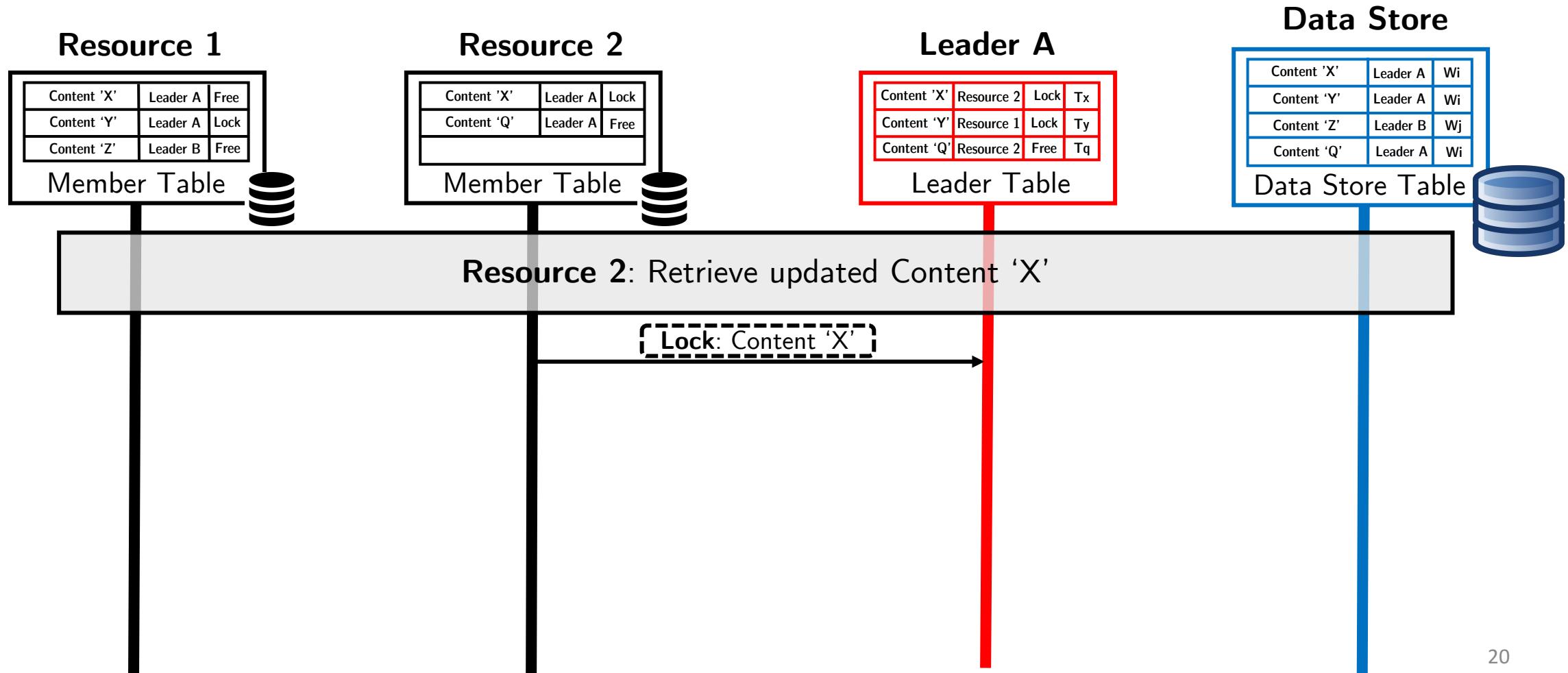
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



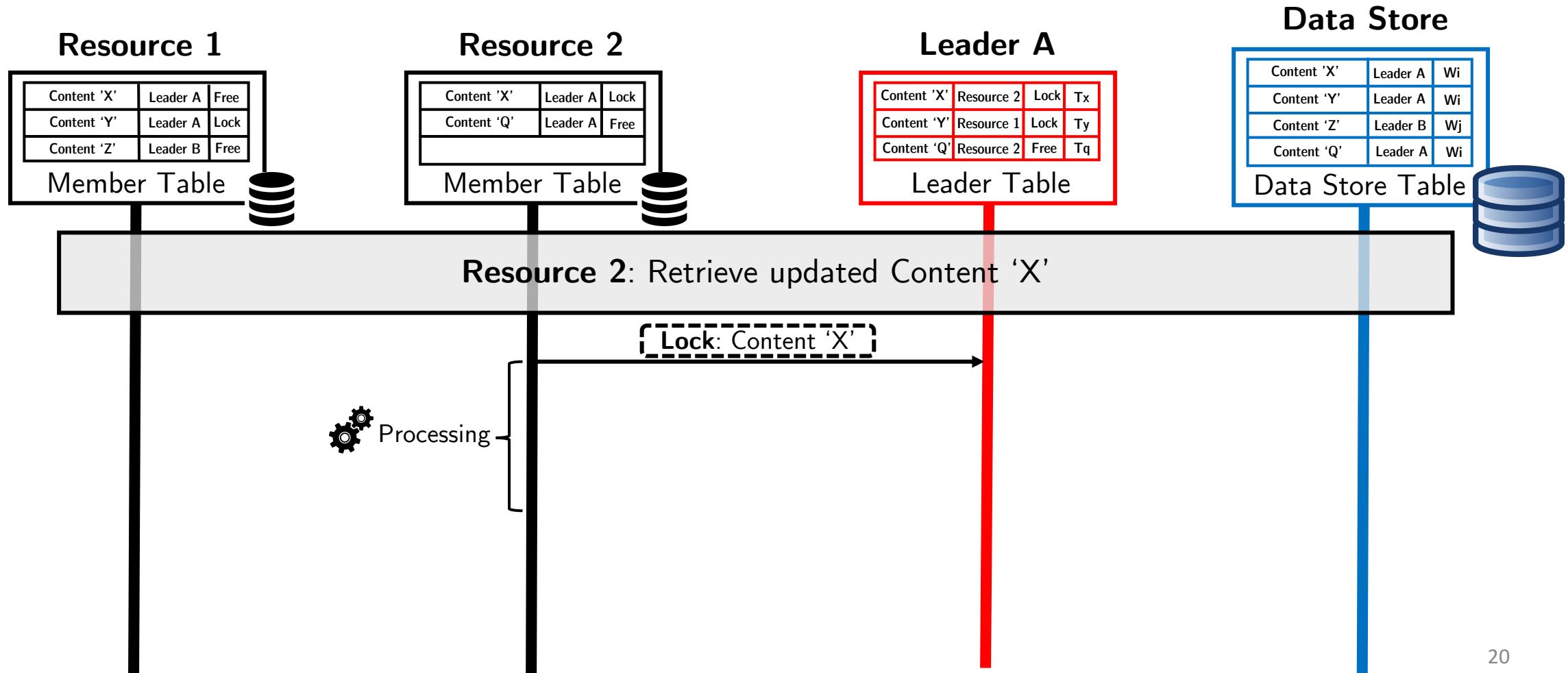
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



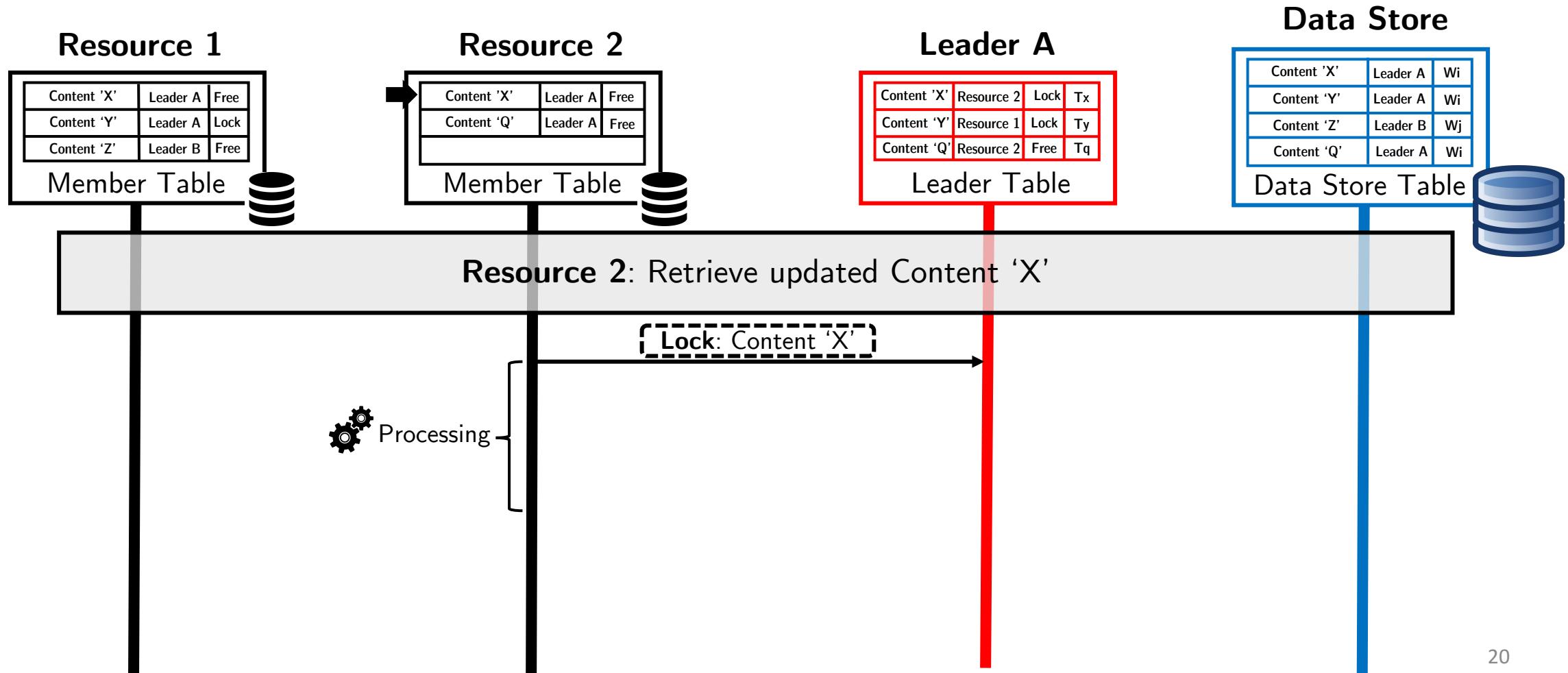
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



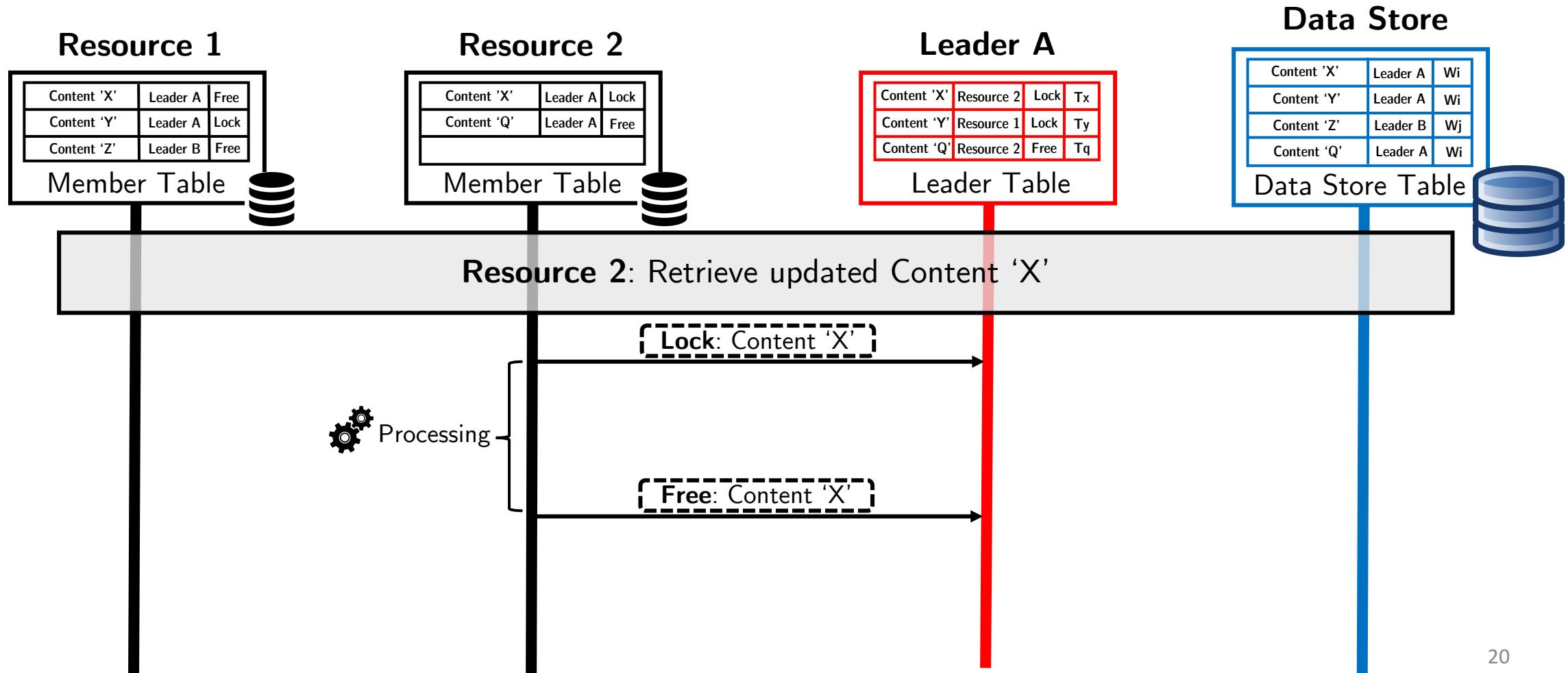
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



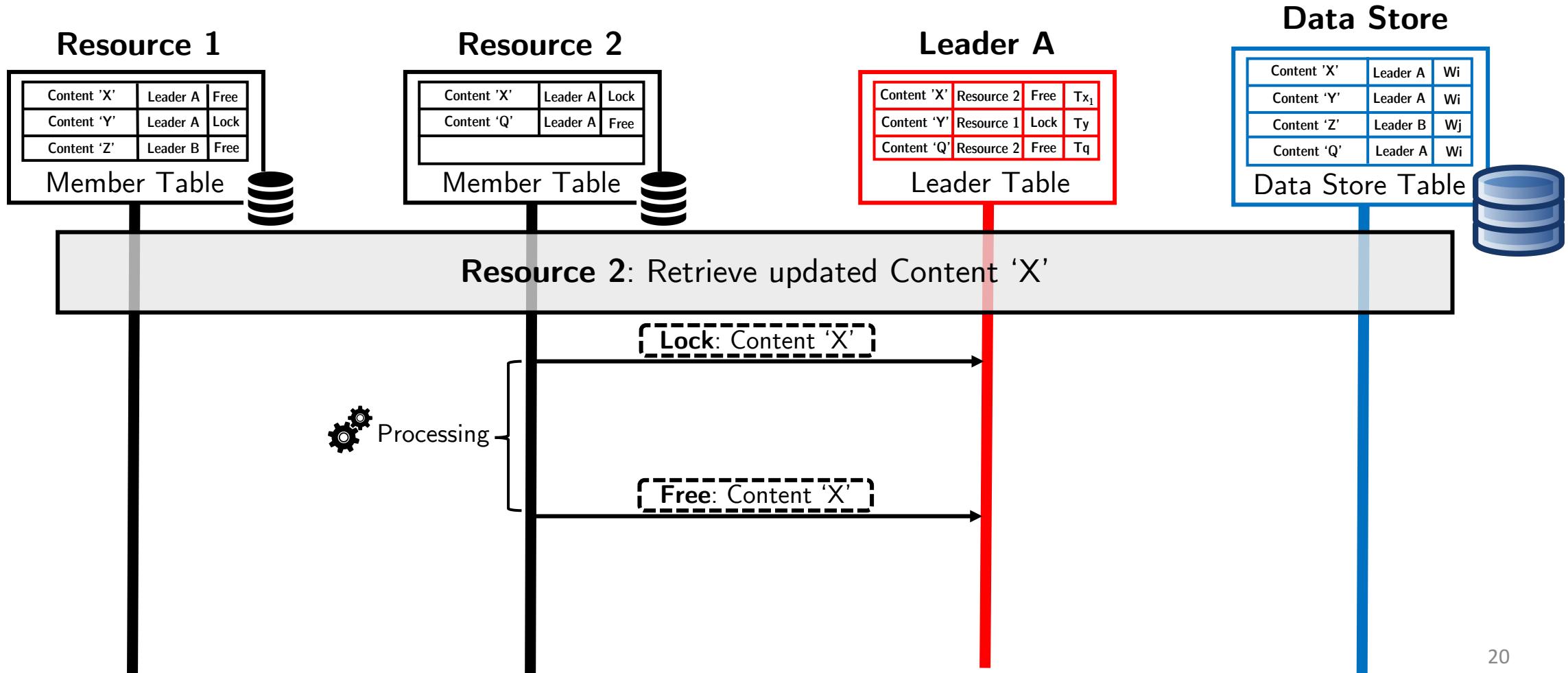
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



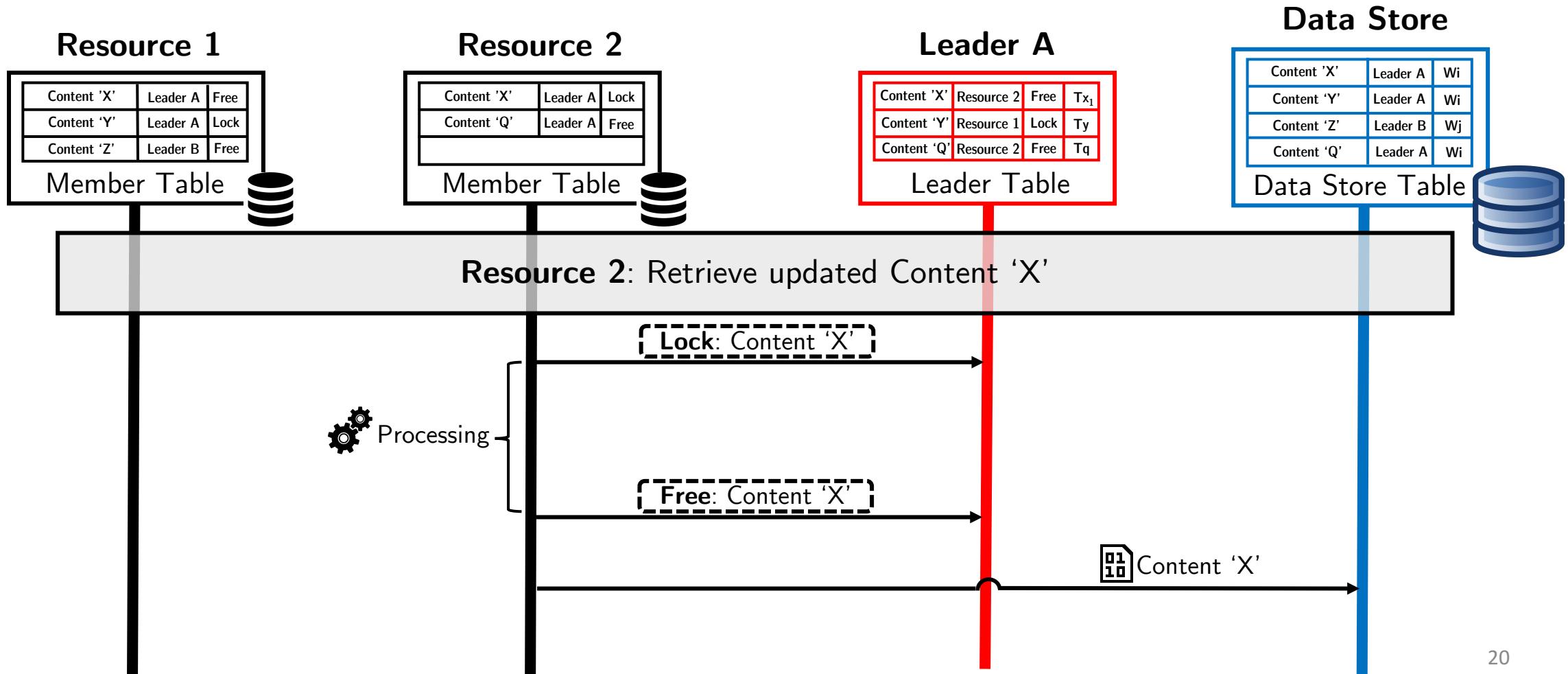
Content Update in Group

Scenario: Update on Content 'X' by Resource 2



Content Update in Group

Scenario: Update on Content 'X' by Resource 2



Requirements

1. Reduce *unnecessary network traffic*
 - Communication model encourages *data sharing* within a group and removes a single point-of-failure
2. Ensure *consistent computation* by avoiding stale data copies
 - Model ensures *causal coherence* on shared data

Evaluation

Evaluation

Evaluated on **Icarus** simulator*

Parameters:

1. 320 Edge and Fog nodes + Data Store
2. Fog node acts as group leader
3. 96×10^4 content items divided in 32 workloads
4. Resource cache size capacity is 10% of overall content size

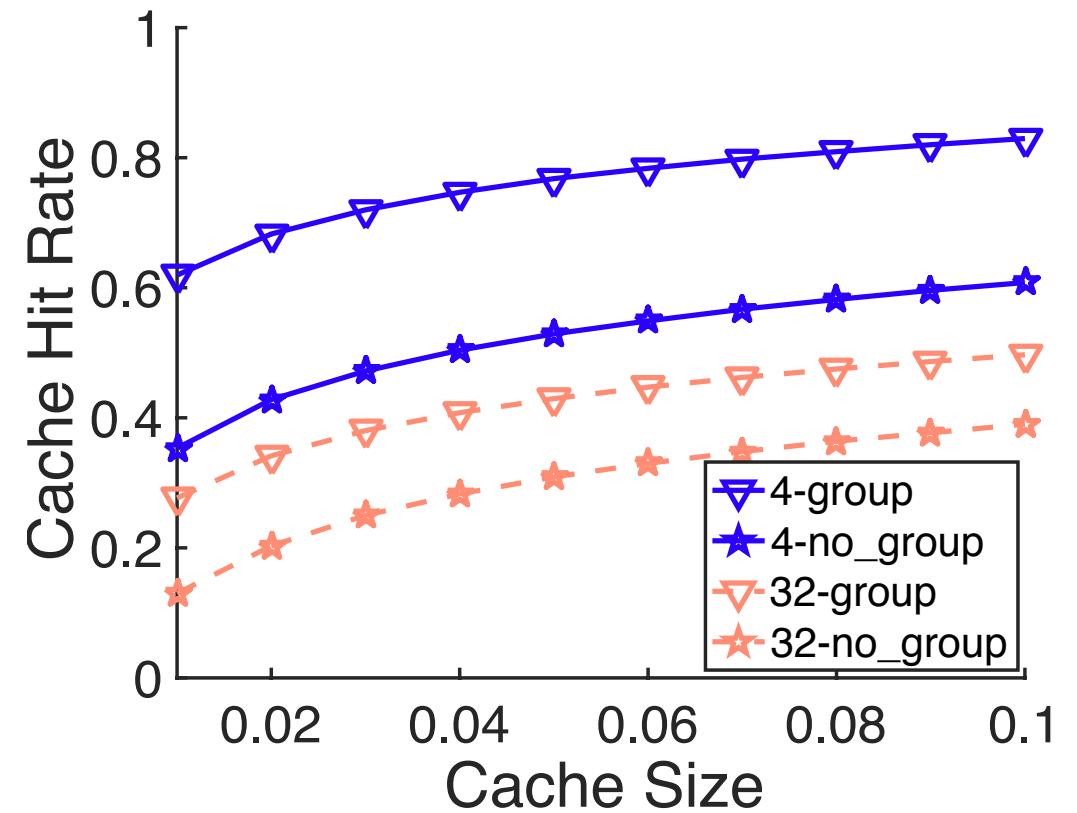
*Lorenzo Saino, Ioannis Psaras, and George Pavlou. "ICARUS: a caching simulator for information centric networking (ICN)." *Proceedings of the 7th International ICST conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

Evaluation

- Q1.** Does resource grouping offer any benefits to edge clouds?
- Q2.** If yes, what is the appropriate size of a group?

Q1. Does resource grouping offer any benefits to edge clouds?

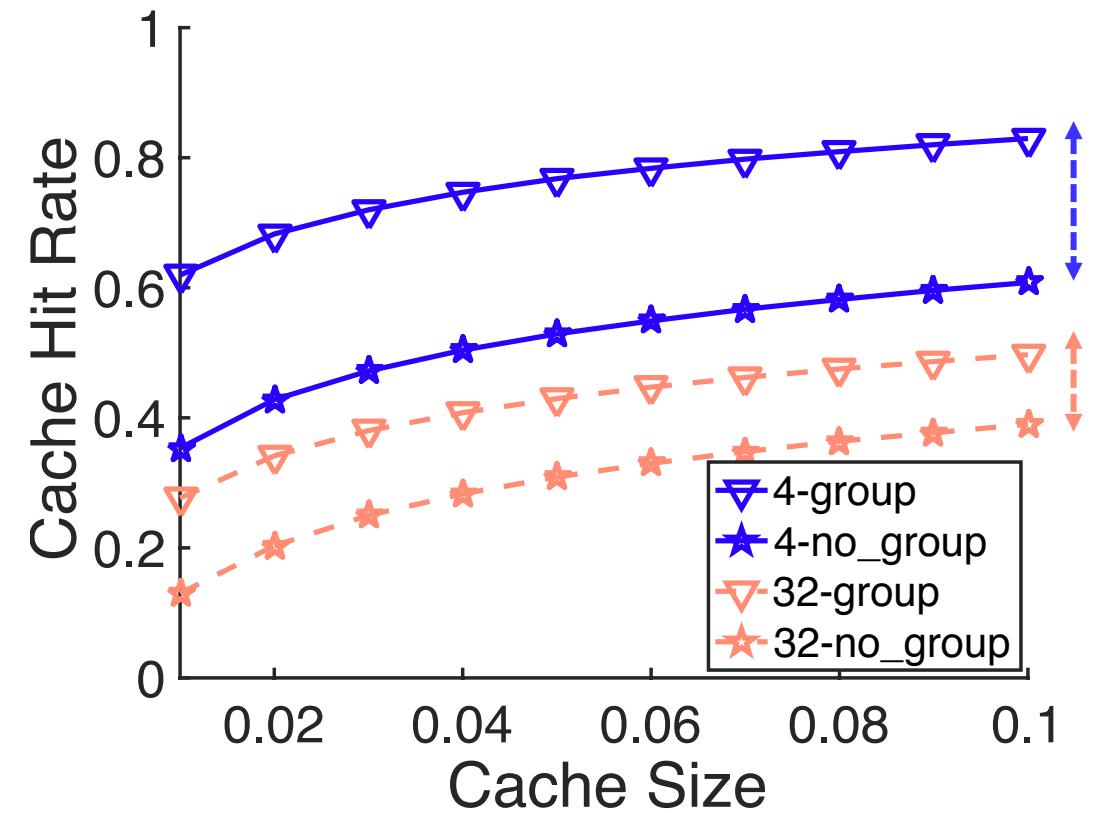
a) Effect on cache hit rate



Q1. Does resource grouping offer any benefits to edge clouds?

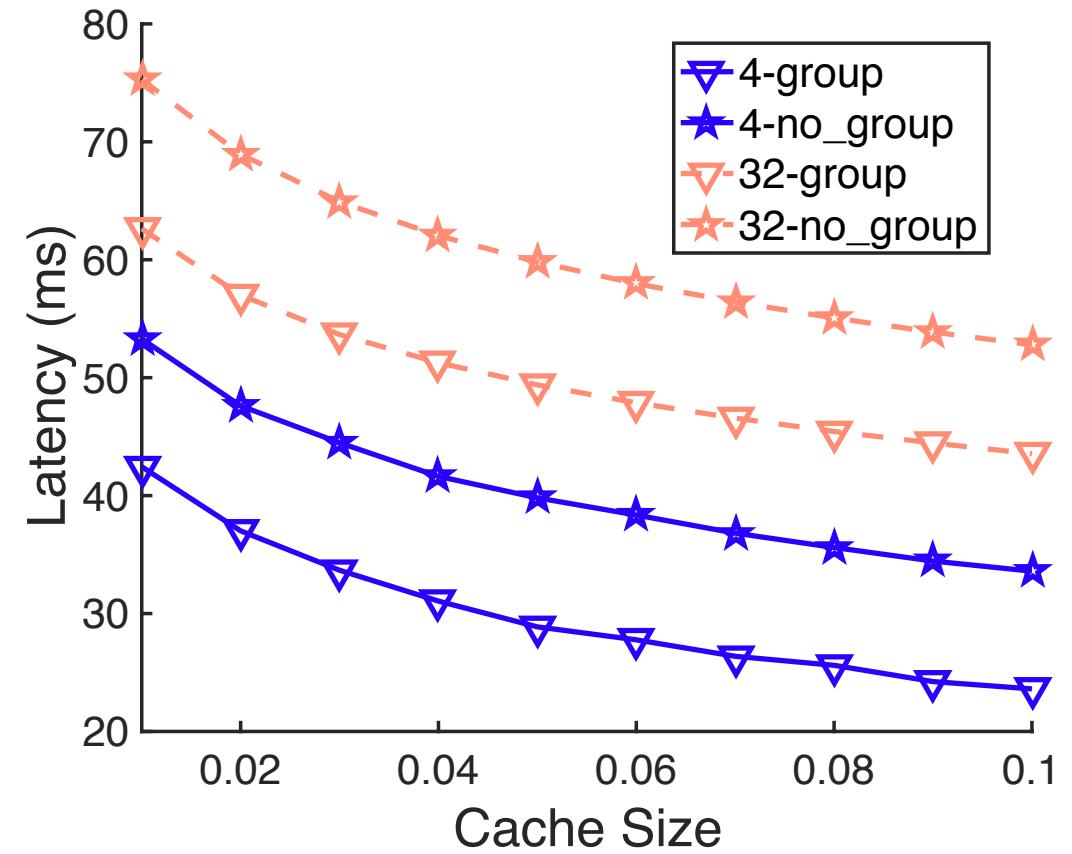
a) Effect on cache hit rate

\approx 2 times increase in hit rate



Q1. Does resource grouping offer any benefits to edge clouds?

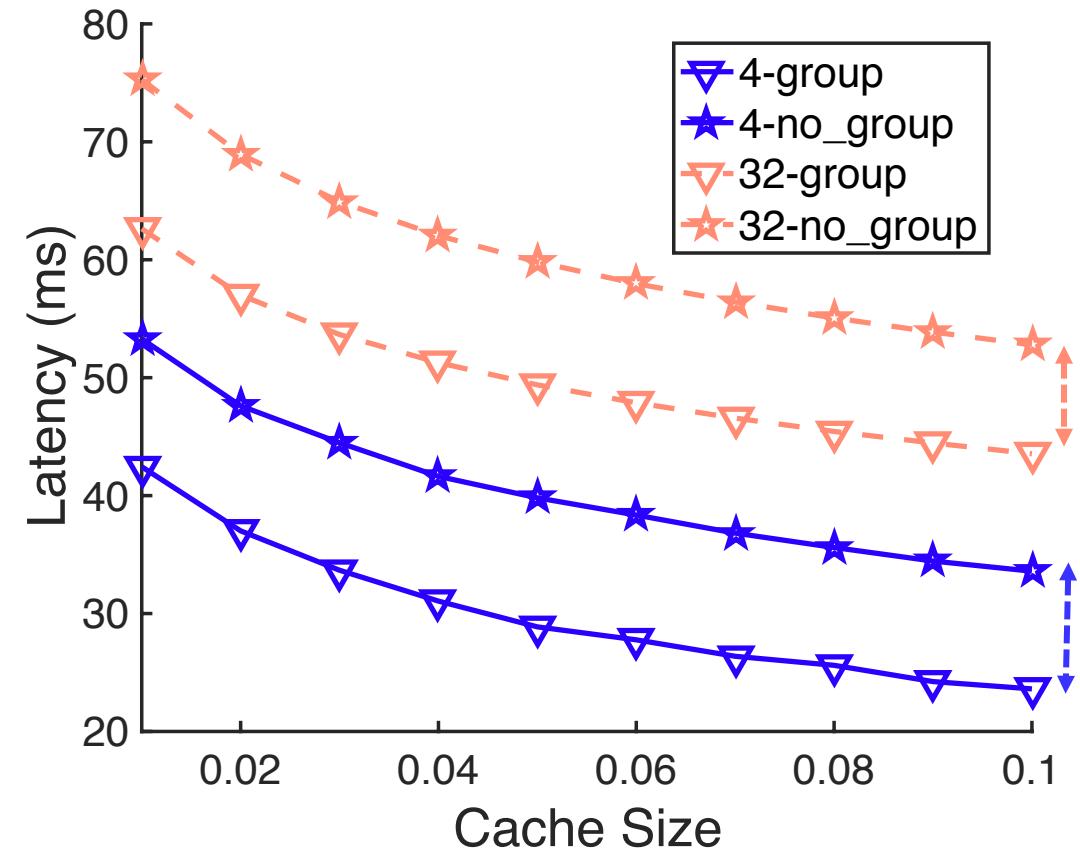
b) Effect on latency



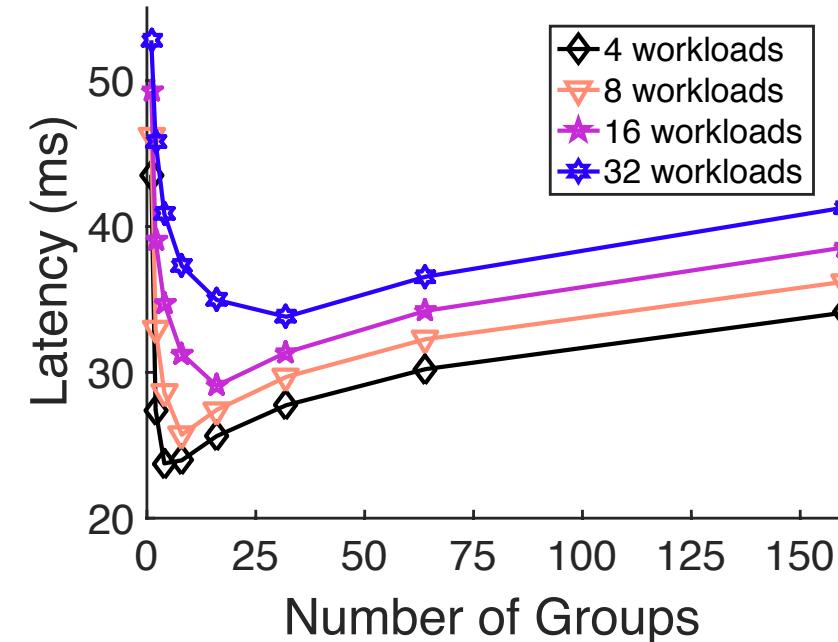
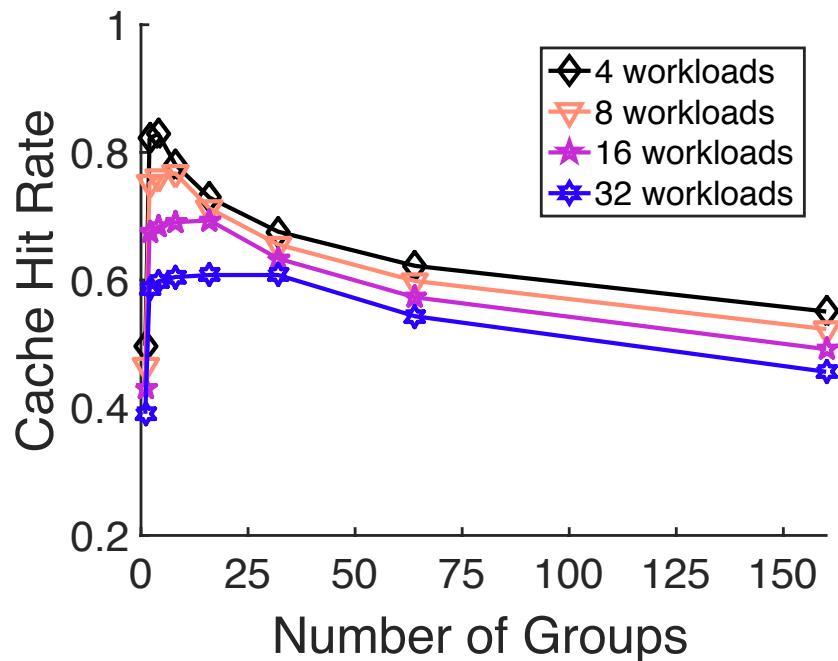
Q1. Does resource grouping offer any benefits to edge clouds?

b) Effect on latency

45% decrease in latency



Q2. If yes, what is the appropriate size of a group?



Maximum gains are achieved when
number of cache groups = number of workloads

Conclusion

1. Edge resource grouping strategy manages content in Edge caches
2. Strategy classifies content based on resource's deployed workloads and cached content
3. The 3-step iterative algorithm maps future deployments on resources with required data locally cached
4. Our evaluations via simulation show
 - i. Direct correlation between grouping and increased system performance
 - ii. Number of cache groups and number of assigned workloads



Thank You! Questions?

nitinder.mohan@helsinki.fi

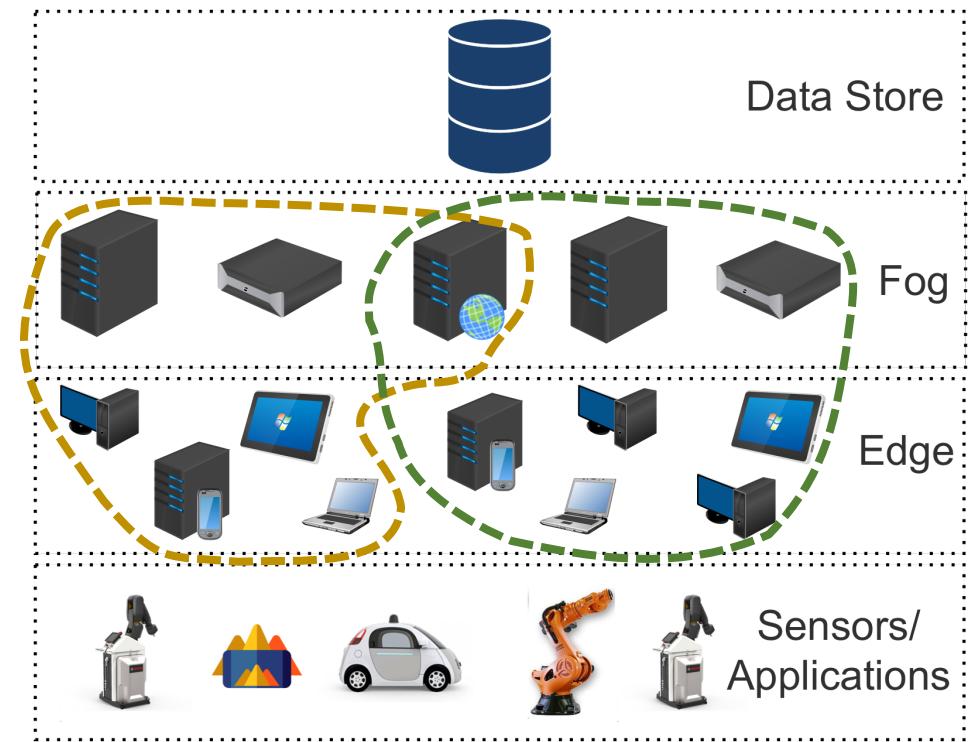


CleanSky - EU FP7 Marie Curie Initial Training Network

Backup Slides

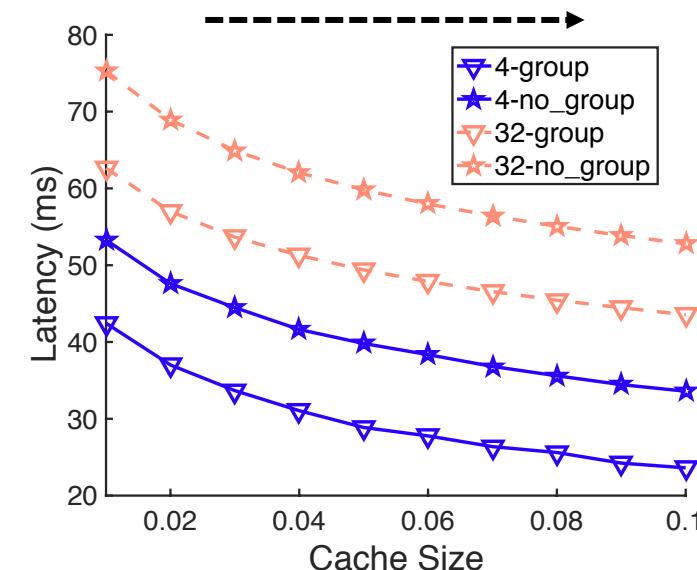
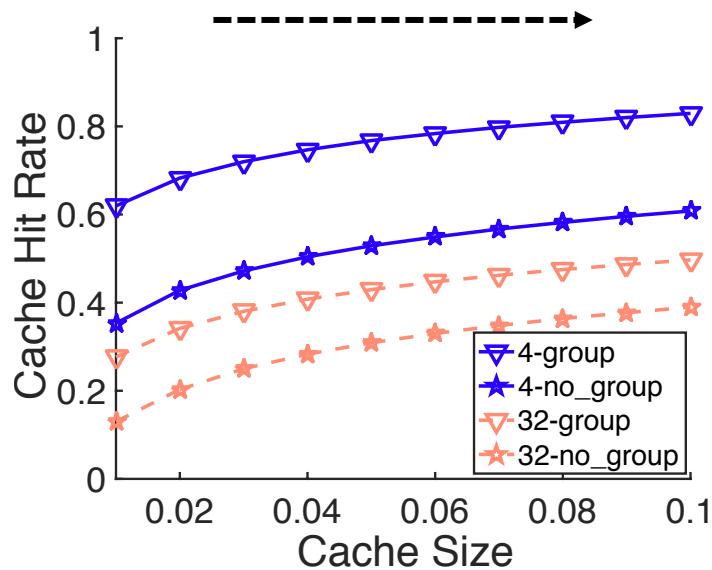
Grouping Classifier

1. **Location:** location of place while constructing AR and 3D models
2. **Relevance:** laptop and cellphone manufacturing tasks as **casing**
3. **Time:** sensor data generation and 'freshness' time
4. **Warning signals:** broadcast of information-only or safety-related signals



Q1. Does resource grouping offer any benefits to edge clouds?

c) Effect of cache size



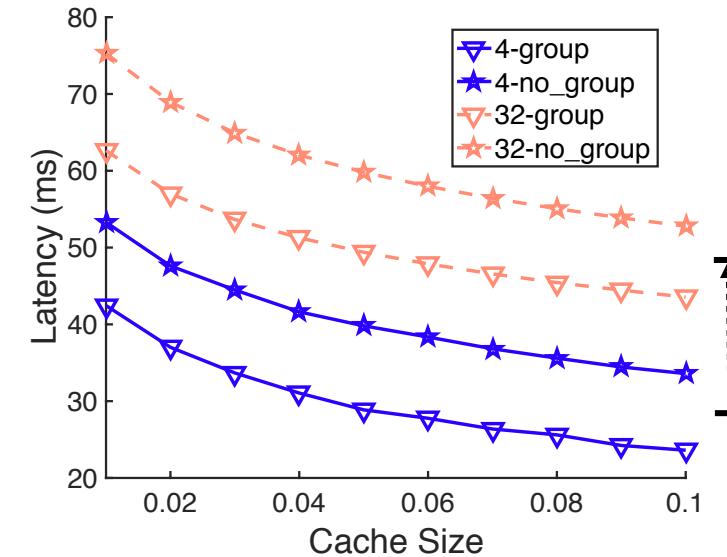
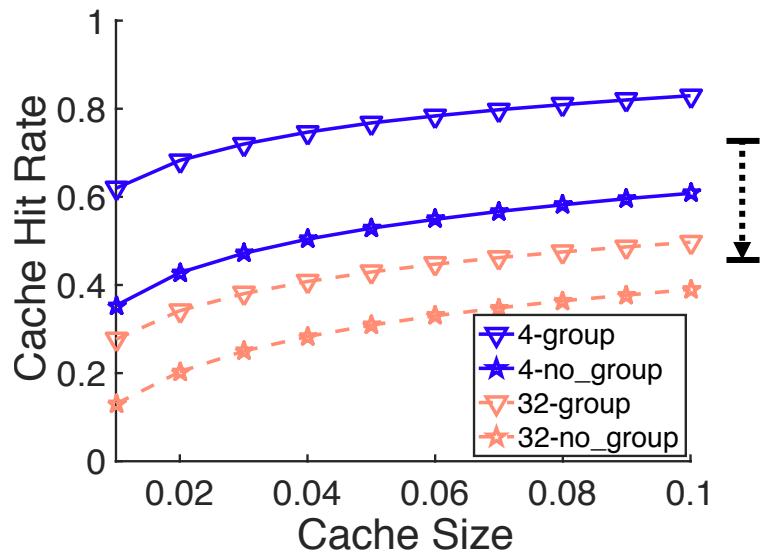
Gains **decrease** as we **increase** cache size!

1. Grouping works best when cache sizes are limited and cannot store all content
2. With larger caches, resources can locally store all content!

Q1. Does resource grouping offer any benefits to edge clouds?

d) Effect of number of workloads

Workloads is a uniform distribution of *similar* content requests



Gains **decrease** on **increasing** number of workloads!

- More workloads \approx more unique content requests which cannot be satisfied by local caches