

Tentative Updates in MINO

Steven Czerwinski

Jeff Pang

Anthony Joseph

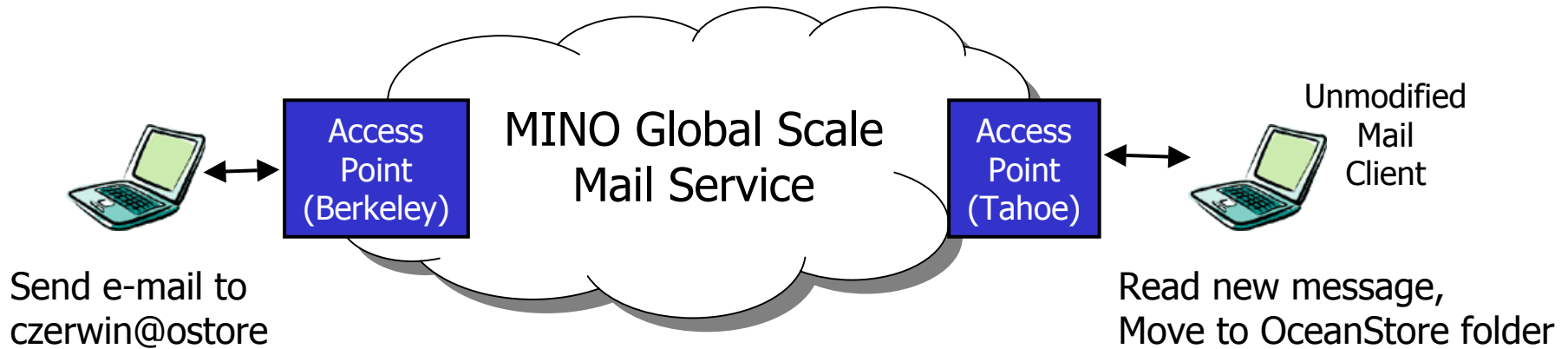
John Kubiatoiwicz



ROC Winter Retreat
January 13, 2002

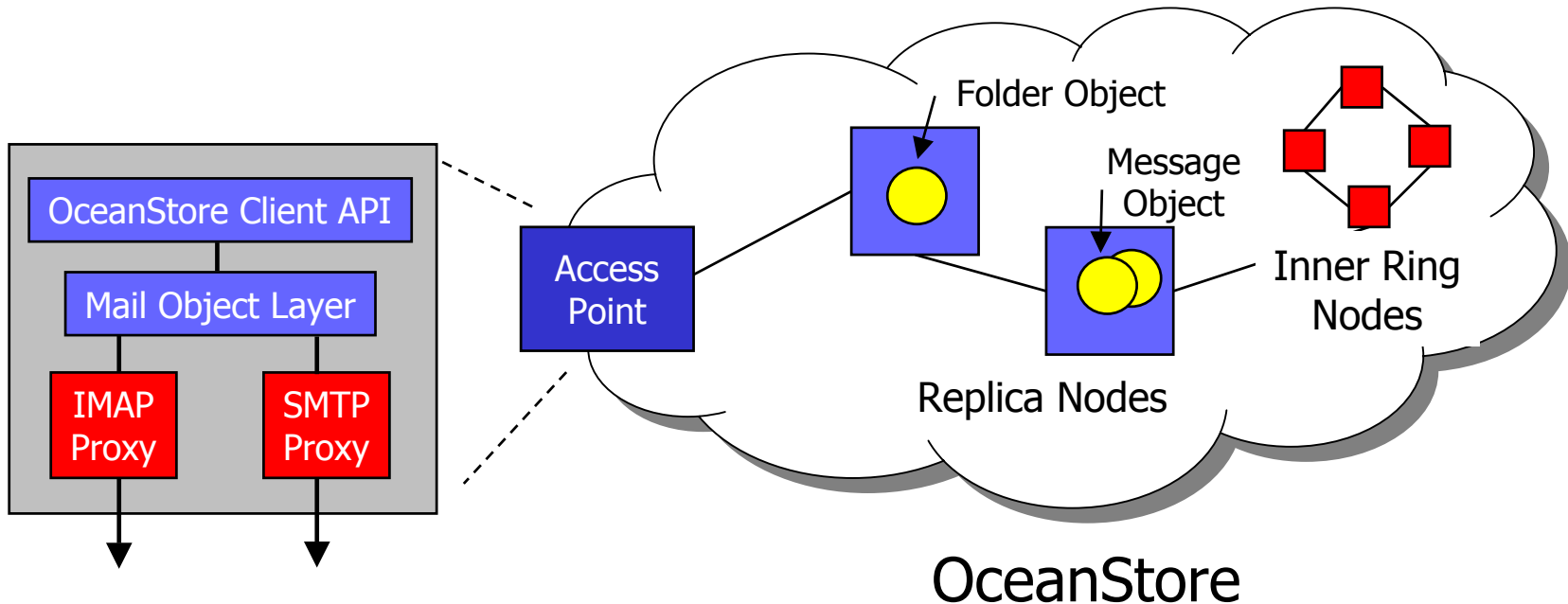


MINO: Mail service on OceanStore



- Enables mail storage and access
 - Individual user accounts with INBOX and folders
 - Send e-mail through SMTP proxy
 - Read and organize e-mail through IMAP proxy
- Interesting application for OceanStore
 - Mail needs global scale and mobility
 - Tested client APIs and tentative updates
 - It's not a file system

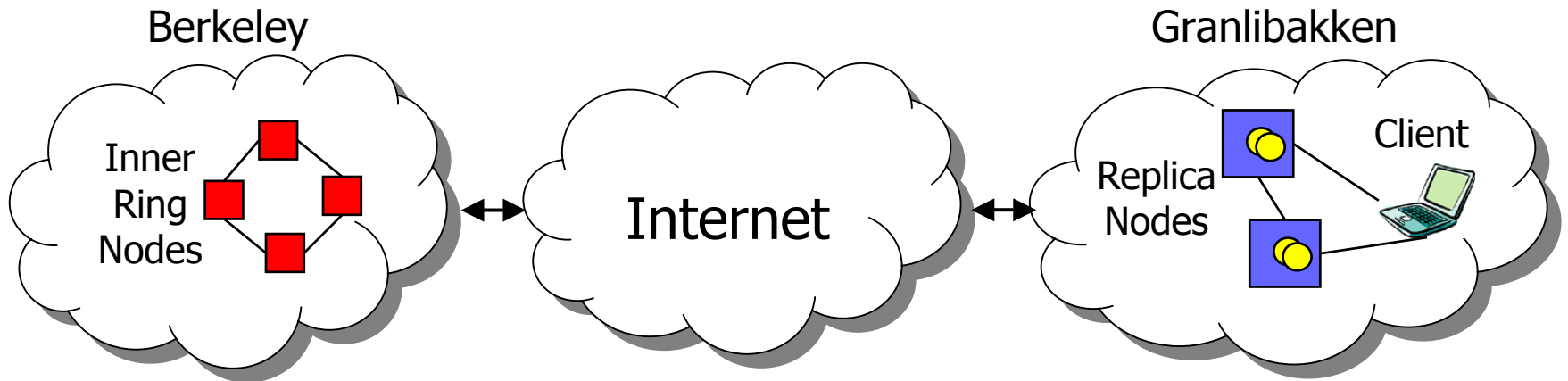
The MINO architecture



- OceanStore provides object store access
 - Inner Ring commits updates, Replicas cache objects
- MINO builds on OceanStore
 - Defines message, folder, mail drop objects
 - Access point is OceanStore client to support legacy protocols
 - Access point typically runs on client machine

Adapting to mobility

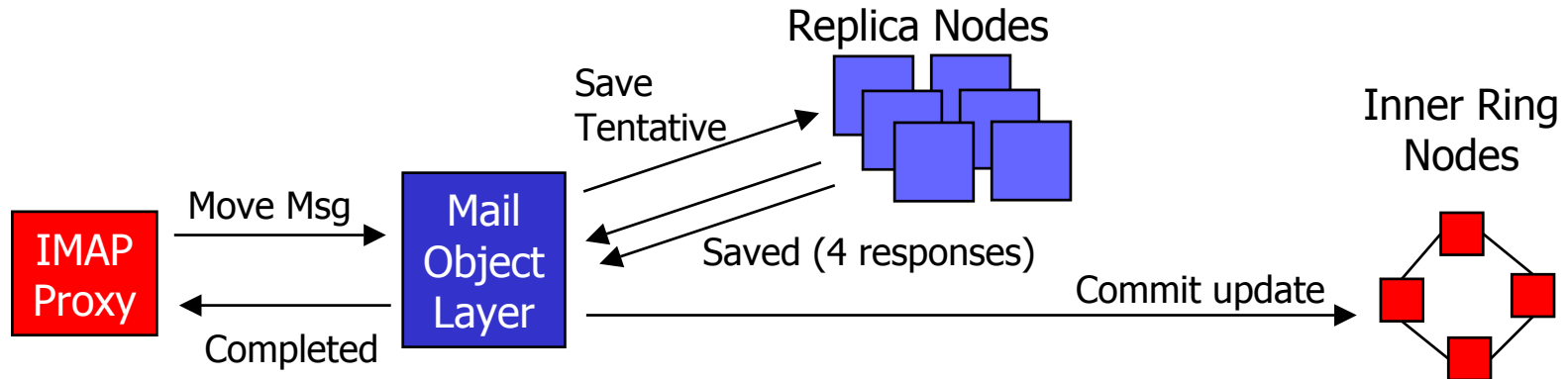
What happens when we go to Tahoe?



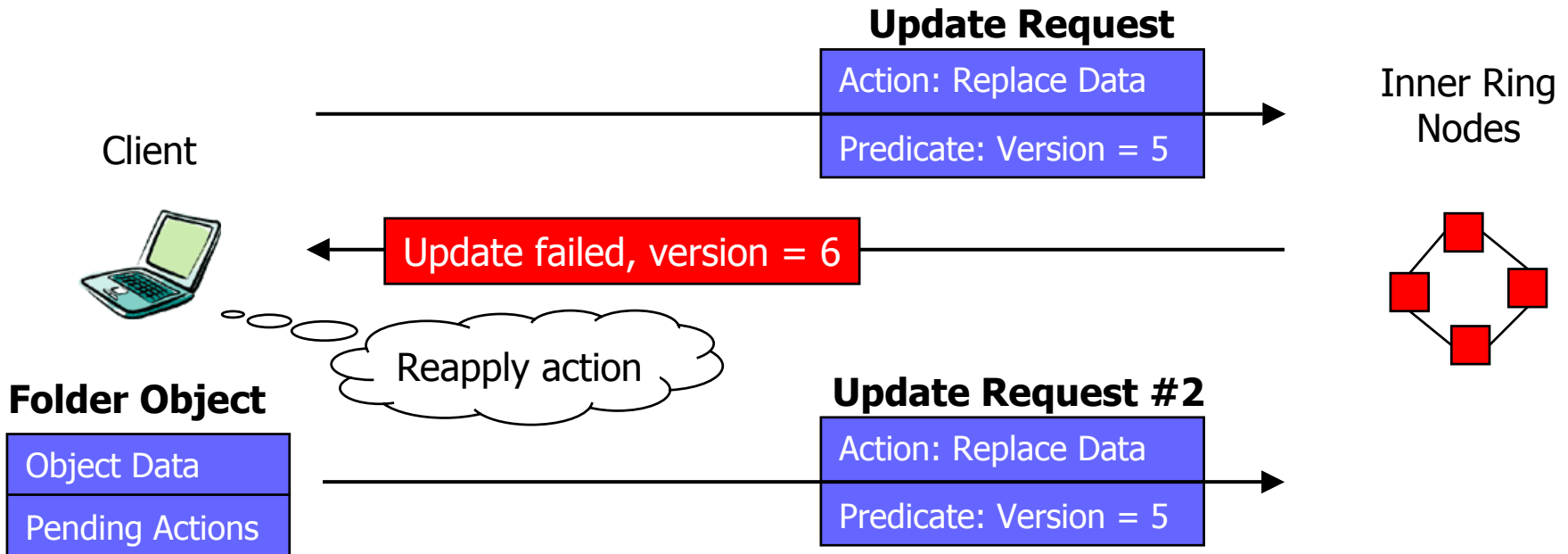
- Replicas of our mail objects migrate towards us
- Updates (writes) must still go to inner ring
- Inner ring is bottle neck to adaptability
 - Can choose replicas based on conditions, but not inner ring
 - Commit time = $T_{\text{transmit to IR}} + T_{\text{in queue}} + T_{\text{process update}}$

Removing the bottleneck

- Use relaxed consistency & tentative updates
 - Don't wait for the inner ring response
 - Consider updates tentatively committed when received by n of m replicas
 - Provides durability
 - Mail layer uses application-specific logic to resolve conflicts

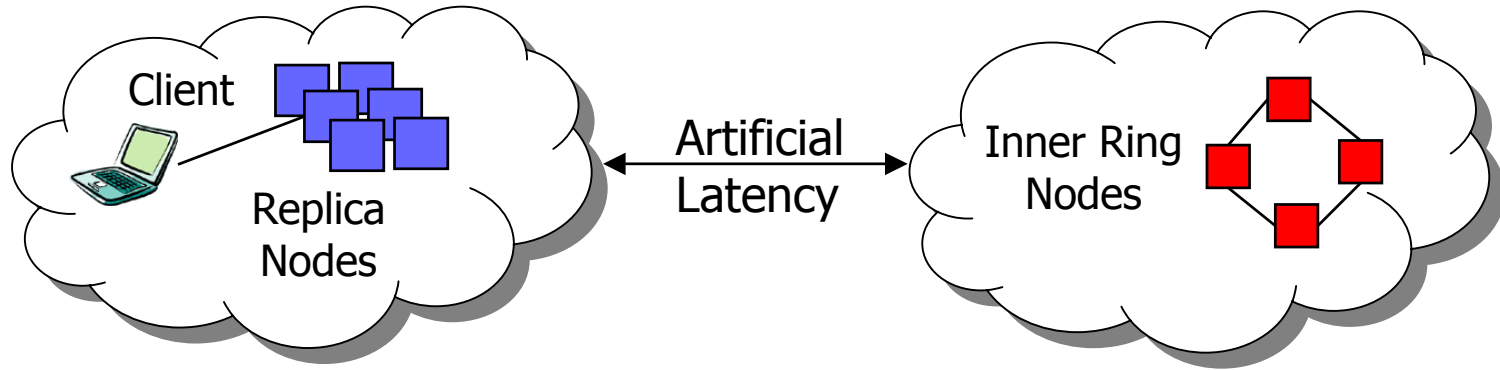


Conflict detection and resolution



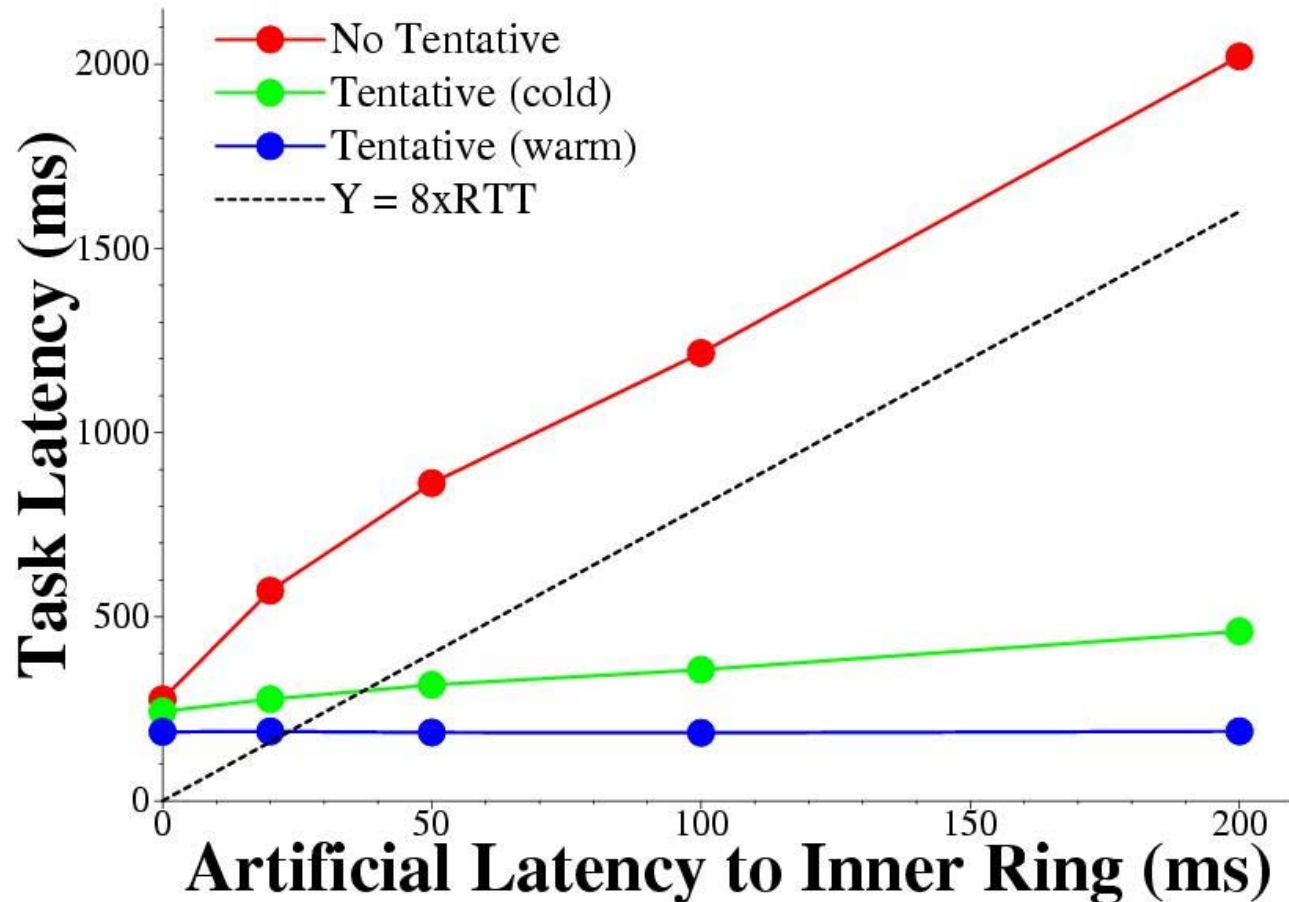
1. Client tracks pending actions
2. Submits update, predicated on version
3. On failure, client is notified and reads newest
4. Checks pending actions for true conflict, then reapplies
5. Resubmits with new predicate

Benchmarking



- Analyzed benefits to mobile clients
- IMAP login benchmark
 - Authentication, client sync, and fetches new msgs
 - 8 blocking IMAP commands
 - Trace from Mozilla 1.0 IMAP client
- Experiment
 - Account with 100 msgs (3 new) and 50 folders
 - 50 trials run for each artificial latency value
 - Cases: no tentative, tentative, tentative w/ warm caches

Results



Tentative updates more robust to increases in latency!

Unresolved issues

- How do we pick remote replicas?
 - Failures should be independent
 - Need at least one well-behaved remote replica
 - Failure rate affects choosing n and m
- How do we know when to use tentative?
 - Tentative benefits when IR slow or far away
 - Puts load on infrastructure resources (replicas)
- What criteria are we trying to optimize?
- Investigating introspection to address issues

Conclusions

- Fully functional mail service with OceanStore
 - Works with unmodified mail clients
 - Supports IMAP and SMTP access
- Investigated tentative commits in OceanStore
 - Client API sufficient to support conflict resolution
 - Showed improvement for mobile clients
- Made suggestions for next prototype
- Future work
 - Using introspection to make optimization decisions
 - Sharing tentative updates with other sessions