Final TripCom prototype

Michael Lafite

WP6 Session
Milano, 8-9 January 2009

Outline

- Implementation progress
- Kernel tests
- Scalability evaluation
- Final kernel
- Open implementation issues

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

Implementation progress

- Optimization of
  - Distributed index storage system (Distribution Manager)
  - Policy evaluation (Security Manager)
  - TS Adapter (→ benchmarks)
- Implementation of
  - Transaction Manager
  - Query Preprocessor (almost finished)
- Support for distributed sub spaces
  - Check whether a sub space is local or not
  - Forward operation to remote kernels, merge results
  - Etc.

New features

- All official API operations implemented:
  - Creation of distributed subspace
  - Support for transactions
- New method `getCatalogue(Space)`
  - Used by query tool developed in WP3
- Creation of root spaces
  - List of root spaces no longer static
  - Multiple root spaces on one kernel
- Synchronous OUT
  - → Simplifies performance testing

New features for use cases

- Non-recursive RD
  - Client decides whether results from subspaces are included or not.
- Completeness information
  - Completeness cannot be guaranteed, but
  - Kernel knows whether a query was complete or not.
- Improved notification mechanism
  - Polling mechanism is hidden
  - Notification via callback
- Single high-level method for creating distributed subspaces
  - No “create remote” for the client

Problems

- Kernel development is ~3 weeks behind schedule
  - Official deadlines (from Helsinki meeting):
    - November 16: all features implemented in the individual components
    - November 30: components tested
    - December 21: kernel integrated and tested
  - Kernel has been in development until the holidays.
- → All features require (further) testing!

Testing

- Make combined effort to test the kernel
  - → Accelerate the process
- All partners run tests targeted at features their work is related to.

Testing

- [FUB+PROFIUM] API methods
  - Test each API method (including subscription, completeness information, etc.)
  - Existing test cases can be reused.
- [CEFRIEL] Security features
- [TUW] Transaction support
- [LFUI (Reto)] Space hierarchies
  - Creation of multiple root spaces, local subspaces and deep space hierarchies
  - RD and RD_NONRECURSIVE in more complex space hierarchies

Testing

- [FUB] Distribution:
  - Creating distributed subspaces, merging results from local and remote kernels, looking up spaces.
- [ONTO+NUIG] Tests with large data sets
  - Start with 1000 triples for each space, then 10000, then 100000, etc.
  - Maybe combined with simple performance tests.

Testing

- [TUW] Run tests with generic scalability test clients.
- [TID, CEFRIEL] Run the use cases on the new kernel.
- [USTUTT] Use the Web Service API.

Testing

- Existing test cases can be reused, updates might be required.
- New tests moved to `kerneltests` directory
  - E.g. API tests in `org.tripcom.kerneltests.api`
- File bugs in bug tracker and provide as much information as possible
  - test case that shows the bug
  - log file
  - Responsible component (if known)

Testing - schedule

- API tests
  - January 15
- Feature tests
  - January 21
- Application tests
  - January 27

- → Kernel ready for scalability tests in February

Scalability evaluation - schedule

- Goal: meet deadlines from Helsinki
- Schedule:
  - January 16: deployment on EC2
  - End of January – Early February: perform measurements
  - End of February: Analyse data
  - March: Document results

Sixth Framework Programme Priority 2
Information Society Technologies (IST)
Specific Targeted Research Project

Final kernel

- New security features:
  - Secure request forwarding
  - Security checks on distributed subspaces
  - Enhanced TAM model
- Tripcom and WSMX integration
- Optimizations

- Not required for final kernel:
  - Query Preprocessor (although probably ready)
  - Self-Organisation features

Final kernel - schedule

- Deadlines from Helsinki:
  - February 1: all features implemented in the individual components
  - February 8: components tested
  - February 22: kernel integrated and tested → final prototype
  - March 15: CD-ROM image, virtual machine images, …

Javadoc

- Every class needs Javadoc documentation
- Only class headers are required
- What kind of information?
  - Short description of what the class does
  - Author? Version? Date? License?

- Exception: TSClient
  - Used by the application programmer
  - → All public methods (with all parameters) will be documented.

Open implementation issues

- Different branch for final kernel
- RD from distributed space not optimized
- Space naming scheme

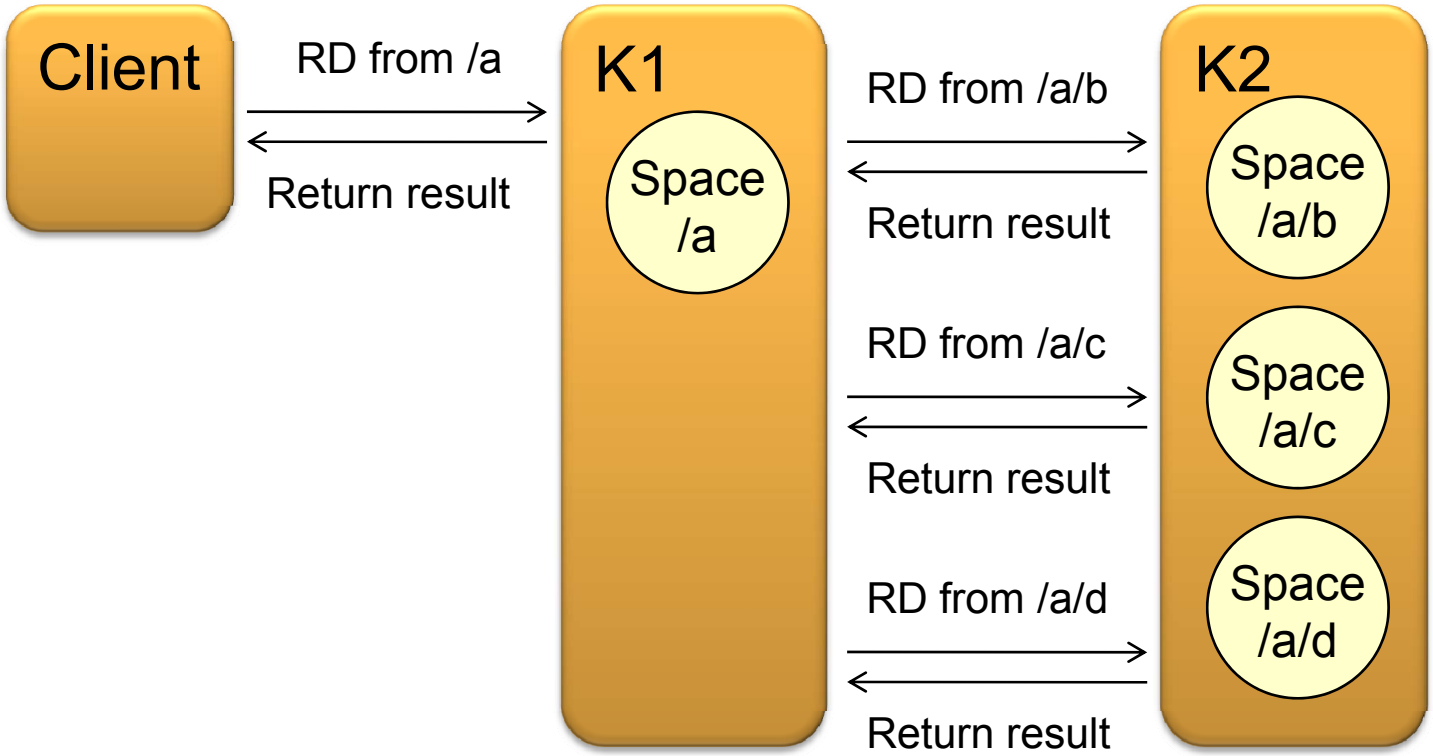
Branch for final kernel

- We need a stable kernel as soon as possible
- Kernel development has to be continued
- →Create branch for final kernel?

- Bug fixes in both branches?

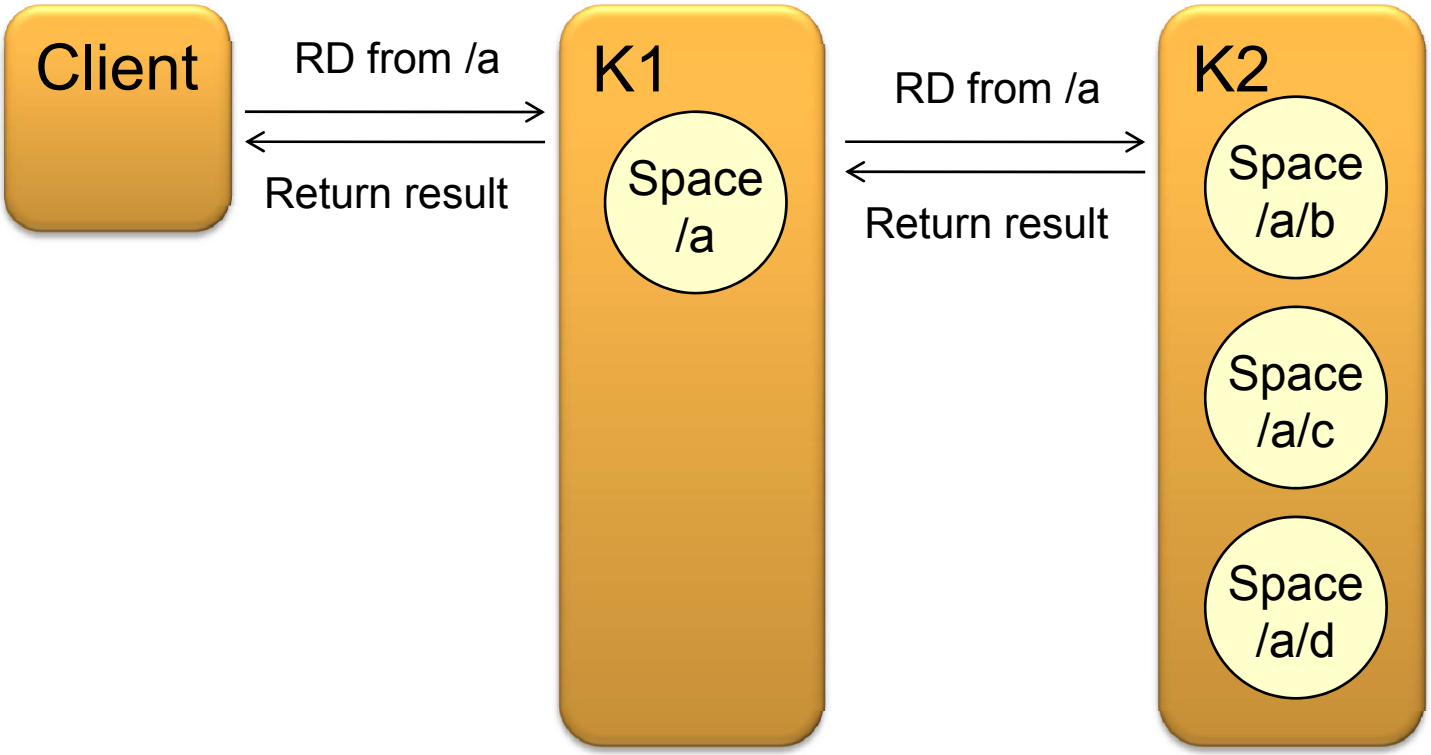
RD - current implementation

- RD from distributed spaces is not optimized
  - Operation is forwarded to the same kernel multiple times.



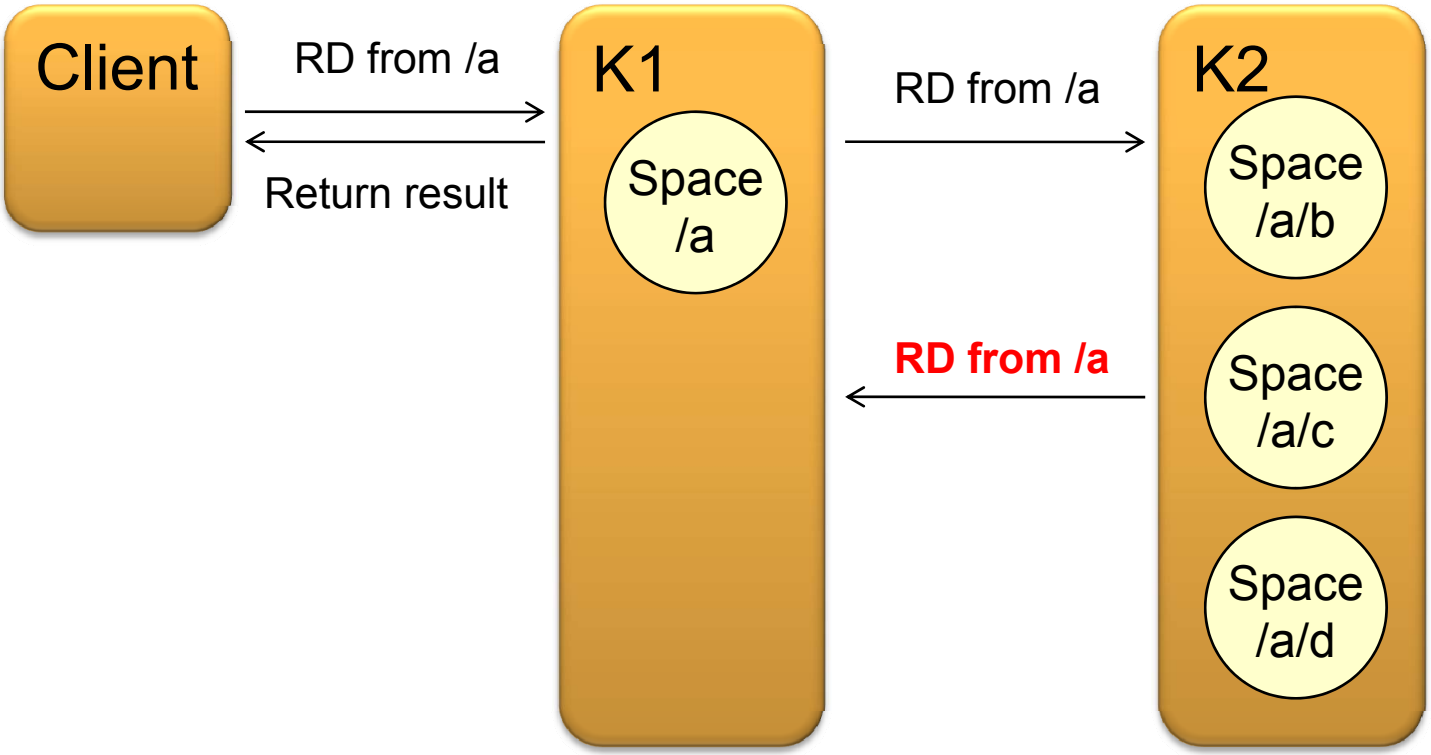
RD – possible optimization

- It should be possible to forward the RD only once, no matter how many direct subspaces there are on the remote kernel.



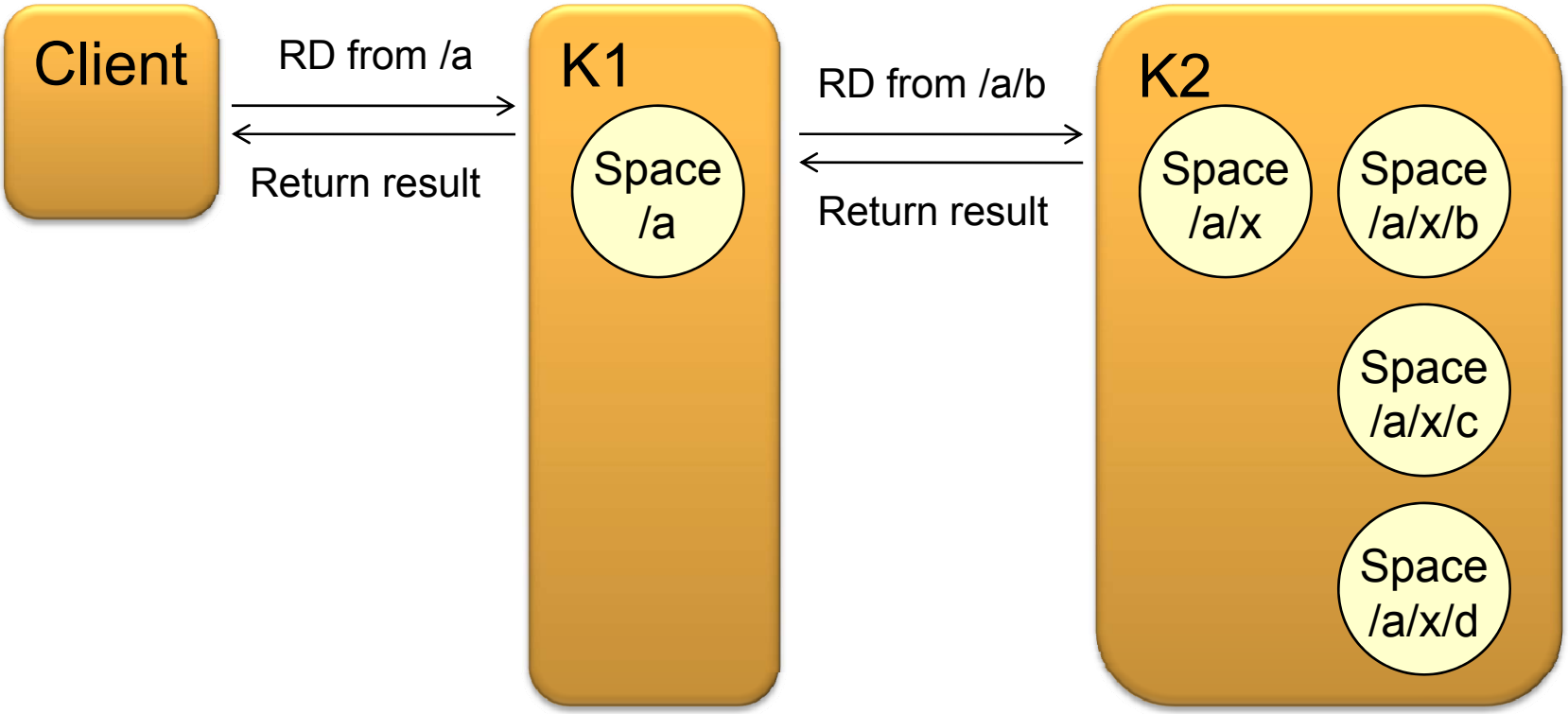
RD – possible optimization (2)

- Problem: K2 will send the request back to K1
- Solution: Tell K2 that it is already executing the recursive RD algorithm.



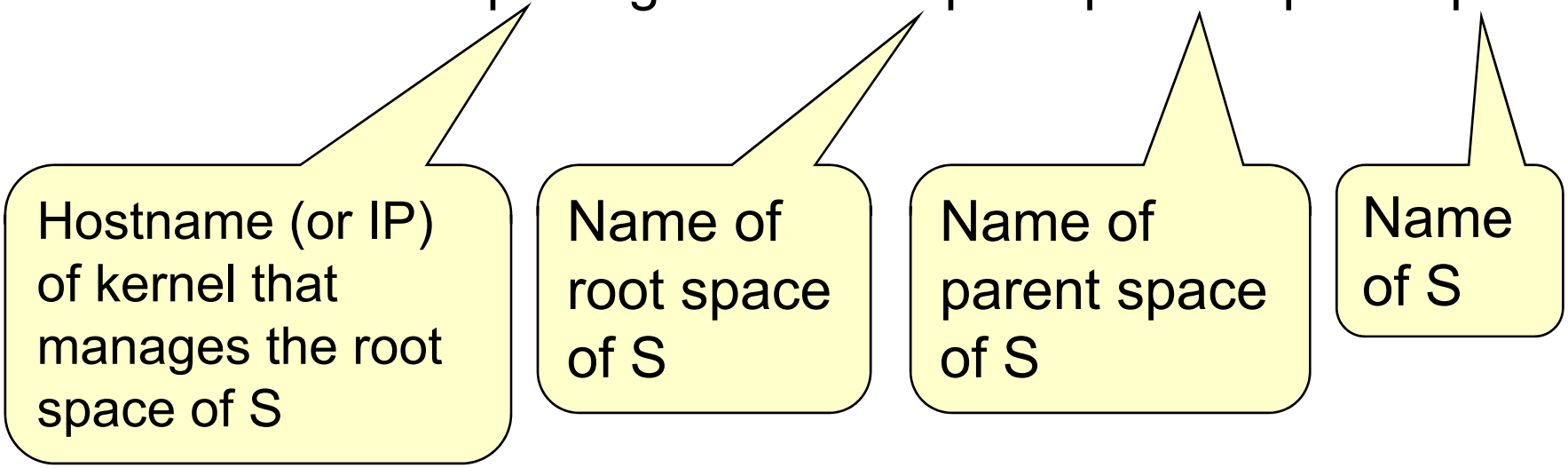
RD – work-around

- Work-around: adjust the space hierarchy



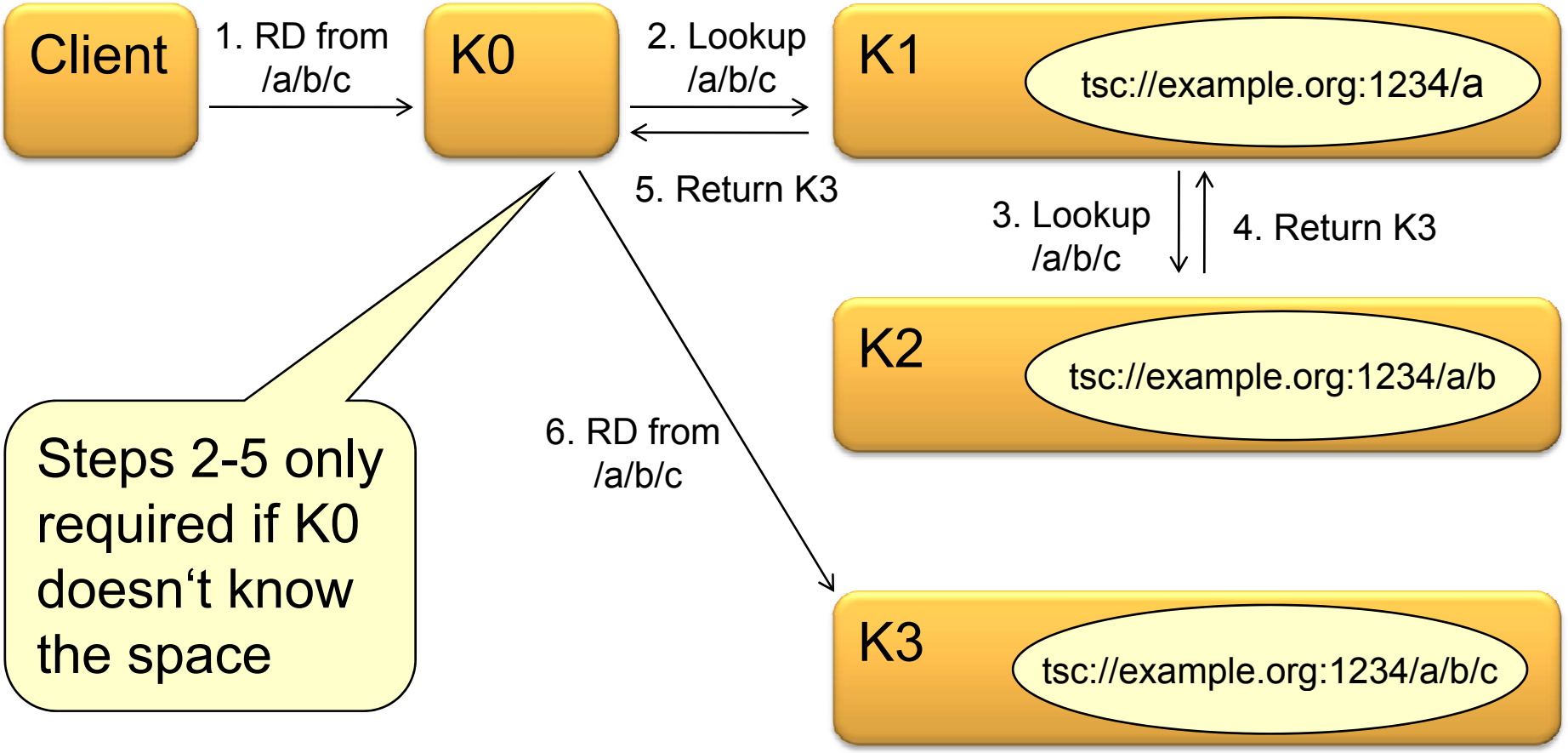
Space naming scheme

- Naming scheme has been changed because of problems related to distributed subspaces.
- New naming scheme:
  - Full space URI of space S = tsc://example.org:1234/rootspace/parentSpace/space



Lookup operation

- Drawback: Spaces now have to be looked up if their location is not already known.



Space naming scheme

- Advantages:
  - Some information about the space hierarchy can simply be inferred from the URI

    → No communication with Metadata Manager
  - No DNS configuration
    - (required for naming scheme based on subdomains: e.g. subspace.rootspace.com)
  - Space not bound to a physical location
    - → Could be moved or replicated.
- Drawback:
  - Lookup overhead

End of Document