Karlsruher Institut für Technologie

Storage CloudSim: A Simulation Environment for Cloud Object Storage Infrastructures

http://github.com/toebbel/StorageCloudSim
tobias.sturm@student.kit.edu, {foud.jrad, achim.streit}@kit.edu

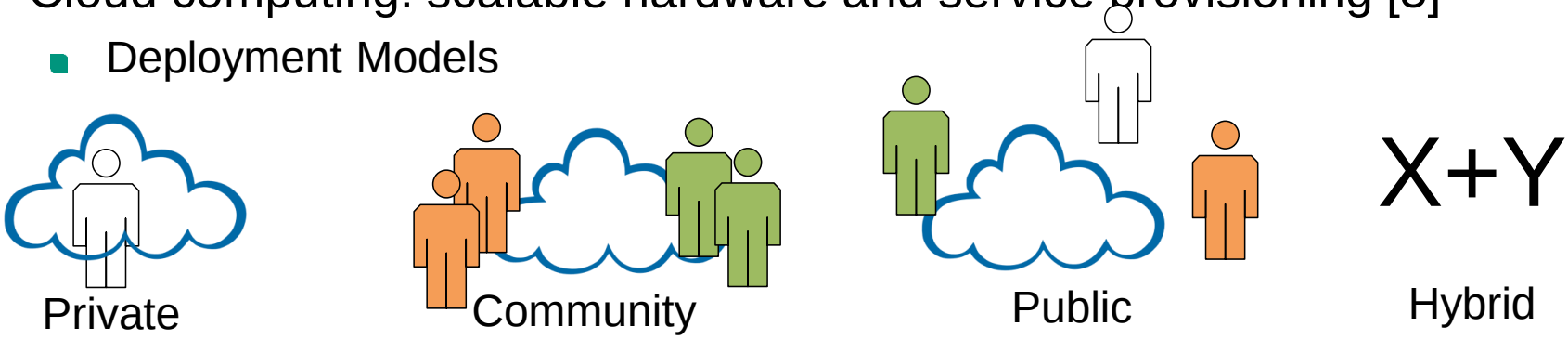
STEINBUCH CENTRE FOR COMPUTING - SCC

Agenda

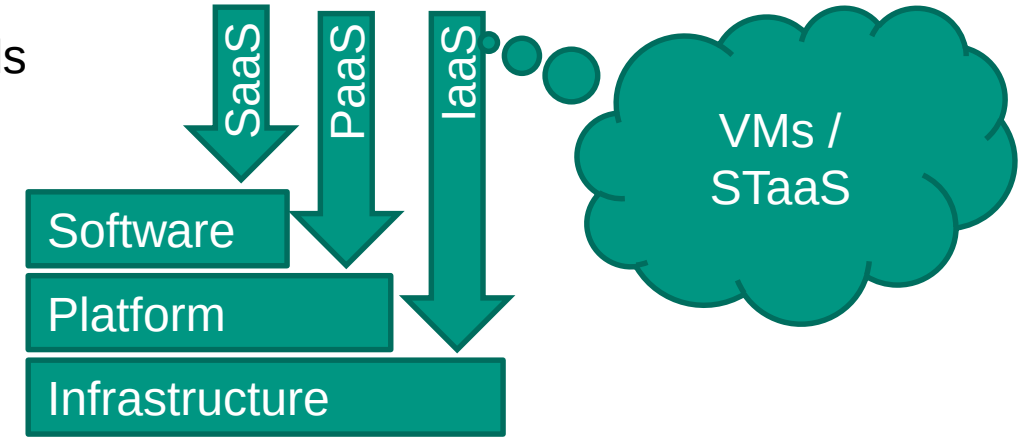
- Introduction & Use Cases
- Motivation
- STaaS Simulation Concept
- Implementation
- Evaluation
- Conclusion & Future Work

Cloud Computing and STaaS

- Cloud computing: scalable hardware and service provisioning [5]
  - Deployment Models



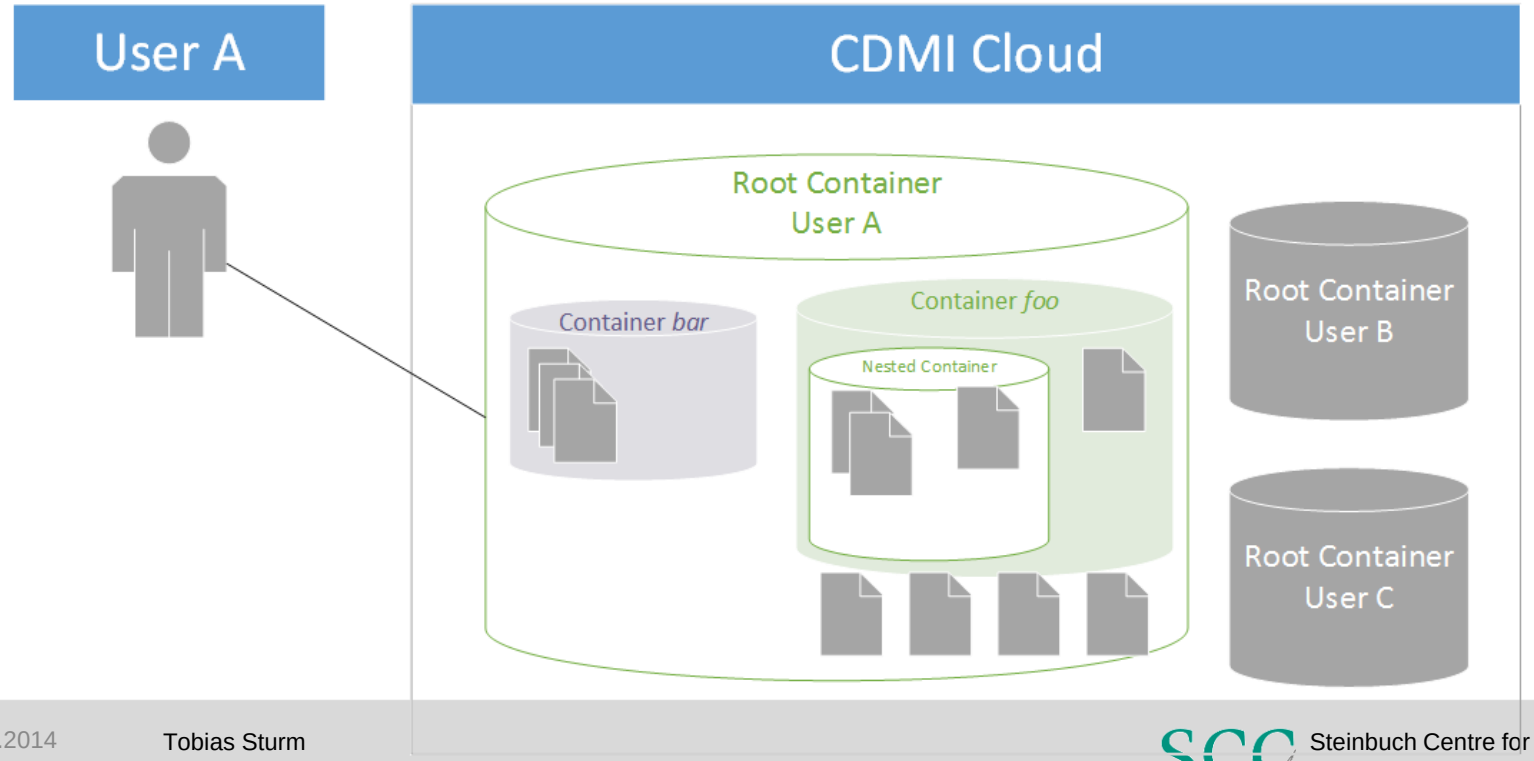
  - Service Models



- Storage as a Service (STaaS)
  - Online / near-line, non-volantine storage at low costs [6]
  - Object or block storage

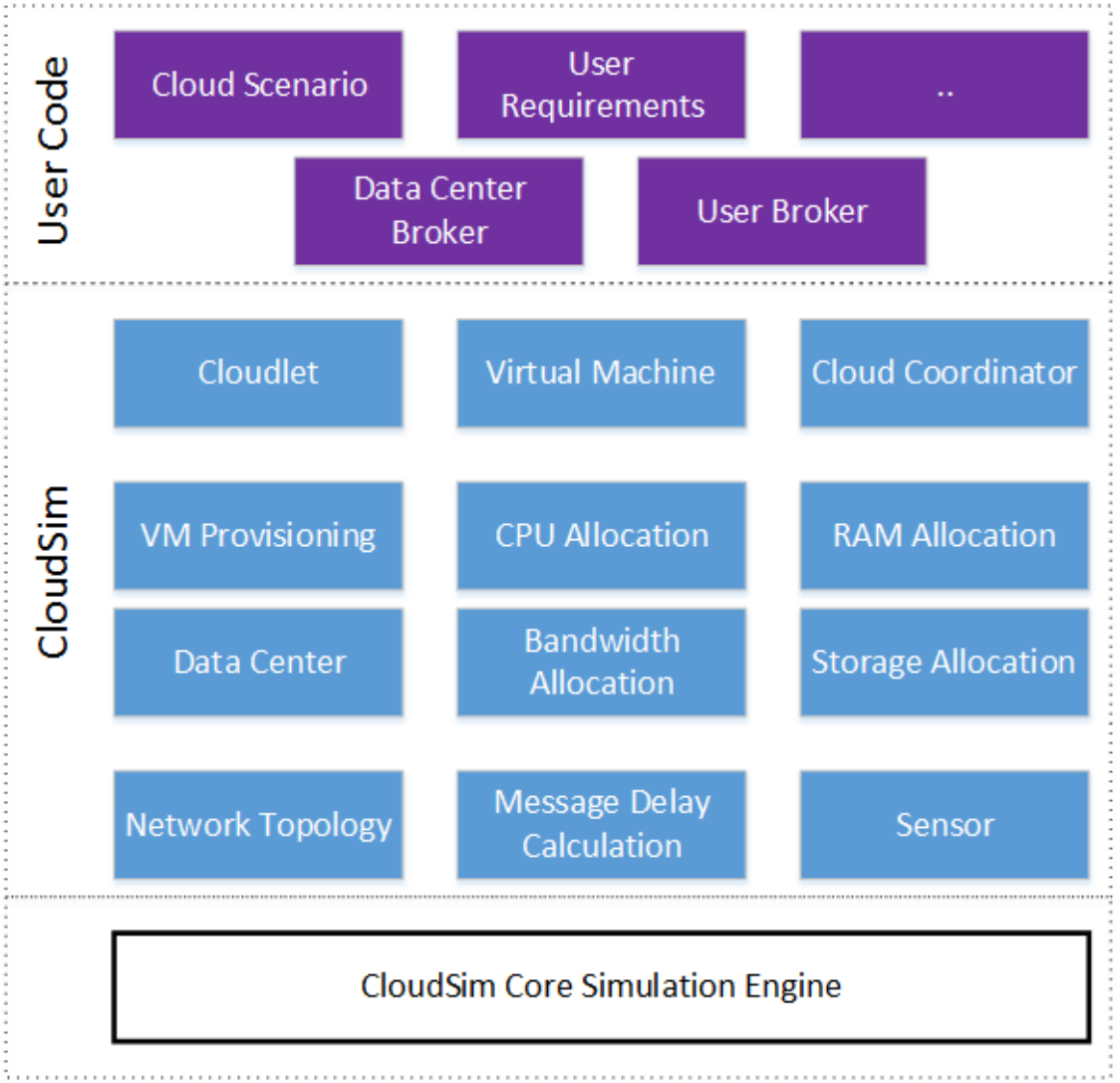
CDMI (*Cloud Data Management Interface*)

- Standard API to access object STaaS, released by *SNIA, 2012* [3]
- RESTful, HTTP based
- Organizes objects in containers, user separation, access via IDs or names



CloudSim

- Popular Java event-based simulation environment for IaaS, developed by CLOUDS Lab at University of Melbourne [1] [2]
- Lacks for STaaS modeling:
  - No standard Cloud user interface
  - Not realistic disk model
  - No simulation of concurrent use of bandwidths
  - Coarse file size model



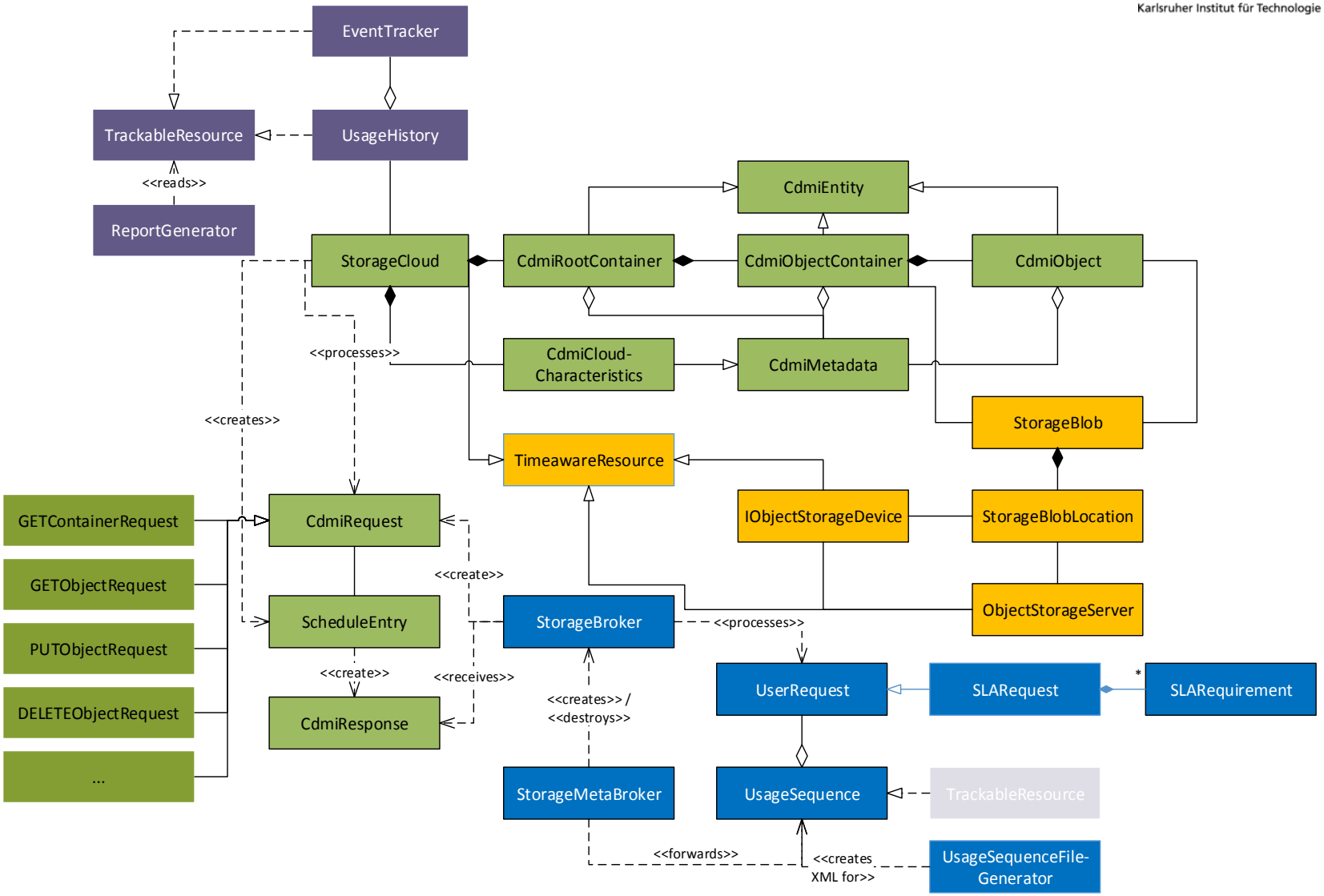
Motivation

- Simulations required to
  - Estimate costs for users in multi cloud scenarios
  - Compare different constellations of hardware components & policies
- No known STaaS simulation environment

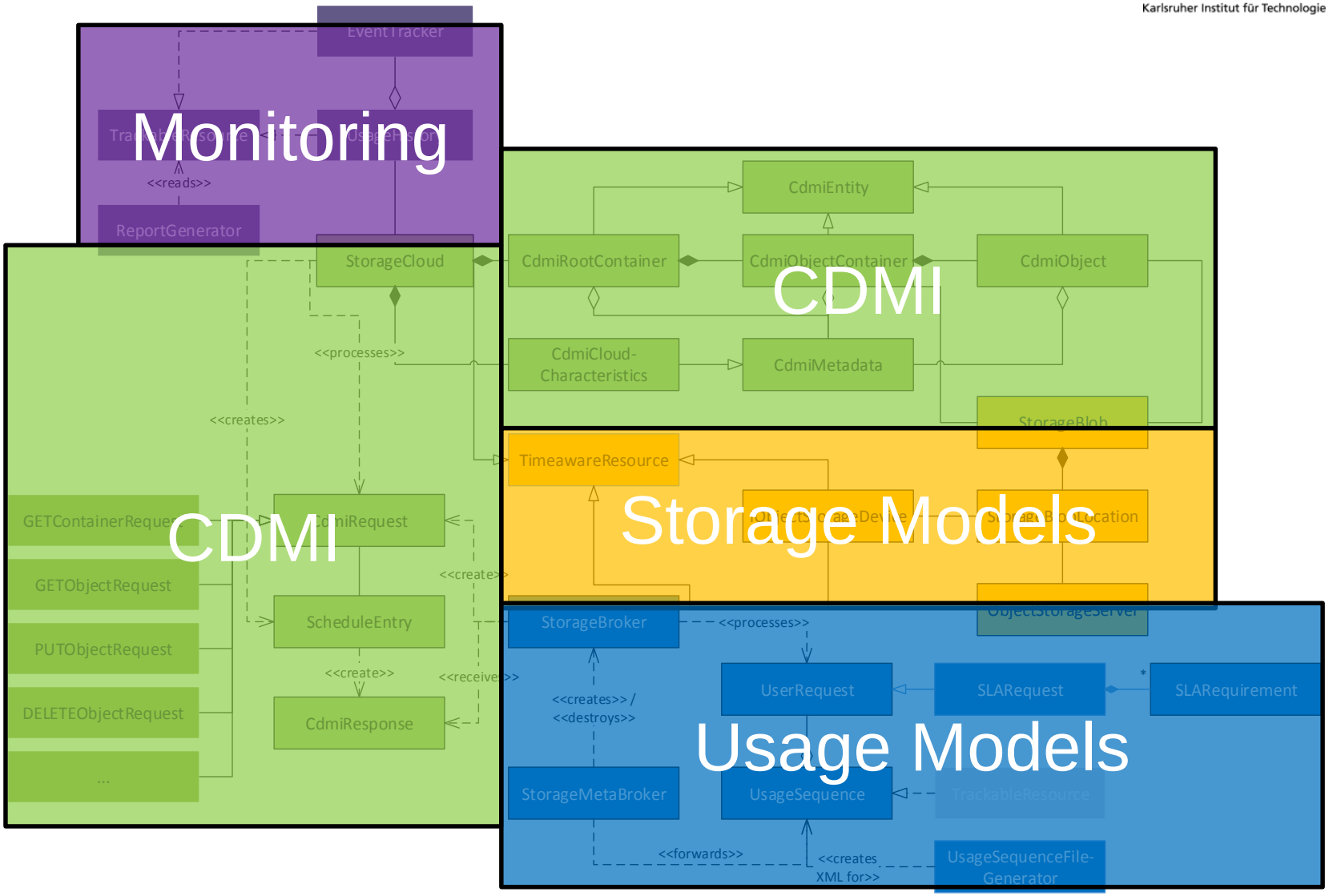
StorageCloudSim

- Develop extension for *CloudSim* to simulate STaaS
  - Accurate models for servers & disks
  - Realistic simulation of IO limitations
  - Use of STaaS standards like CDMI
  - Model multi-Cloud STaaS usage

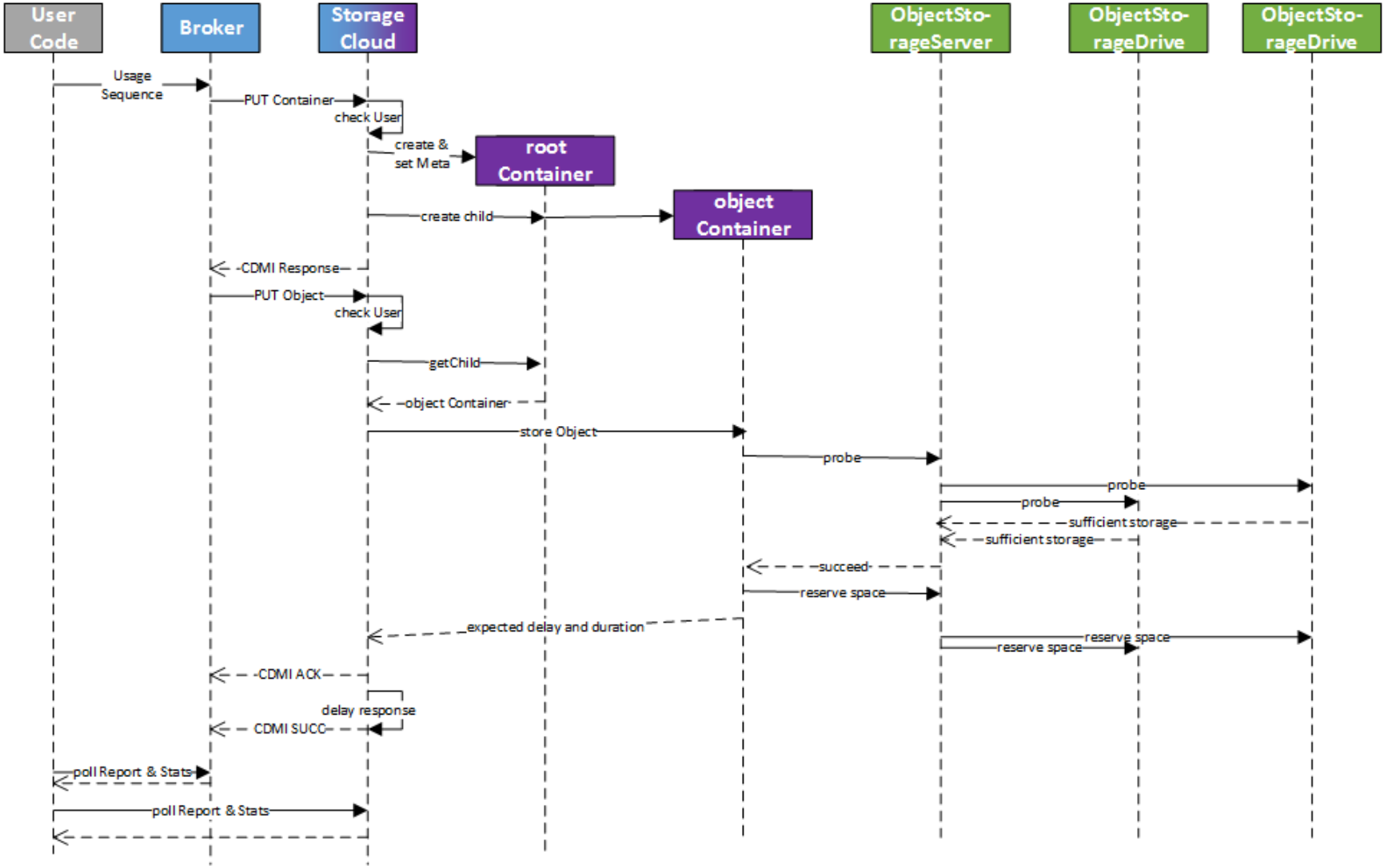
Implementation - Class Diagram



Implementation - Components



Example STaaS Request Workflow (Single Cloud)



Implementation - SLA based StaaS Brokering

- SLA requirements are defined in `UsageSquence`
- `UsageSquence` independent from each other
- `MetaBroker` chooses best provider for each `UsageSquence` using SLA matching policies and SLA rankings
- Example SLA matching policies:
  - Supports Capability (WebDav export, metadata modification, …)
  - Does not have Restriction (max. container/object size)
  - Max./min. allowed feature metric (max. latency, min bandwidth, max storage cots per GB,…)
- Example SLA Ranking: Assign Score to each Cloud:
  - $score = \frac{const}{price\ per\ stored\ GB}$
  - Rate availability of capabilities

Evaluation – STaaS Clouds Setup

- Modelling of a single- (Amazon S3) and multi-Cloud (all three) scenario
- Linear pricing model

	Amazon S3	SCC intra	Swift Cloud
#servers / #disks per server	6/6	1/3	4/4
write/read rate, write/read latency capacity per disk	156 MB/s 9.5 ms / 8.5 ms 2 TB	64 MB/s / 156 MB/s 11 ms / 9 ms 1 TB	156 MB/s 9.5 ms / 8.5 ms 2 TB
# allowed replica	3	3	1
Max. obj. size	Unlimited	16 GB	unlimited
total capacity	72 TB	3 TB	32 TB
$ per uploaded GB	0.0002 $		
$ per stored GB $ per down. GB	0.05 $ 0.01 $	0.04 $ 0.0002 $	0.01 $ 0.1 $

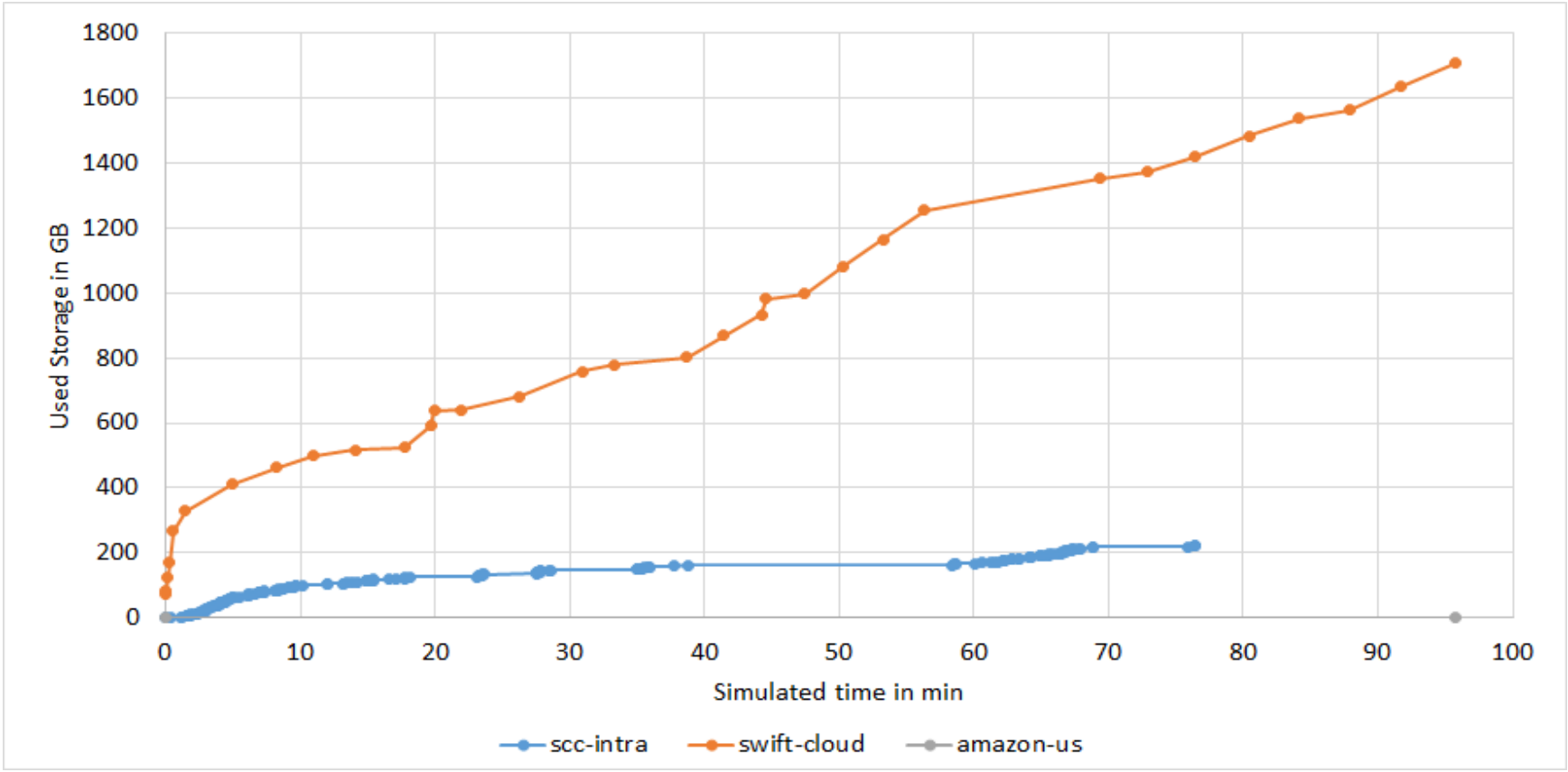
Evaluation – Modeled STaaS `UsageSquences`

- Simulations with two types of `UsageSquences`
- Three experiments: mixed (50, 500, 5000 input sequences), normal only and scientific only (250 input sequences)

	Sequence A	Sequence B
total traffic per sequence	Gamma distr. $\alpha = 2, \beta = 3$, max 15 GB	1..100 GB (uniform distr.)
upload-download ratio	3:1	uploads only
size of biggest object	1 KB .. 1 GB	1..100 GB (uniform distr.)
Idle time between two requests	10 ms – 30 s	Bursts of 5 operations, 5-10 min idle between bursts
SLAs	cdmi_create_container cdmi_delete_container SLA available capacity > Y	cdmi_create_container cdmi_delete_container No max_container_size max_object_size > X SLA available capacity > Y
Rating	$\frac{1}{store} + \frac{1}{upload} + \frac{1}{download}$	$\frac{1}{store} + \frac{1}{upload}$

Evaluation – Effect of Object Size

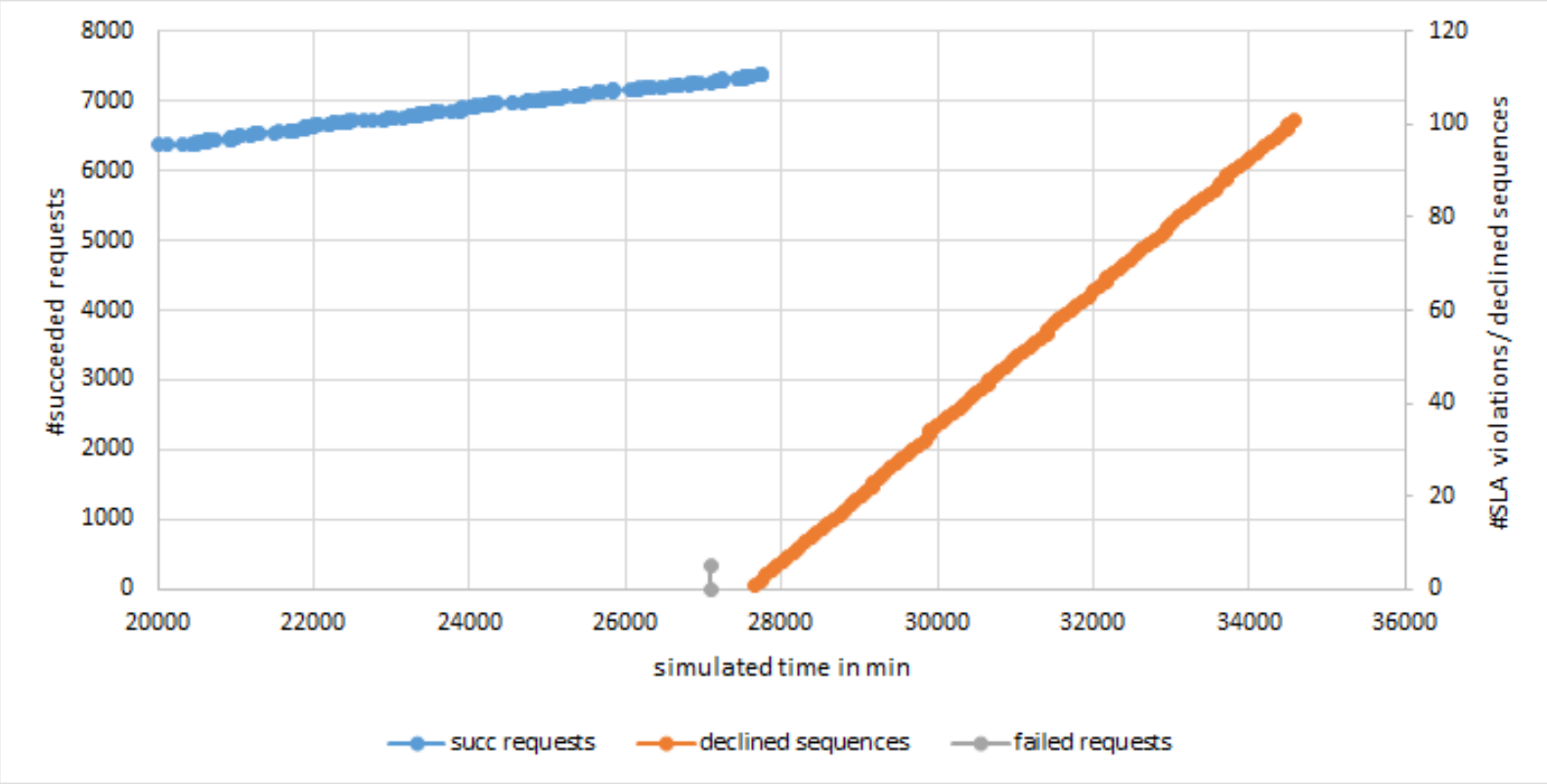
- `MetaBroker` selects Clouds with lowest Costs with respect of SLA
- SCC Cloud selected for small objects as it is the cheapest Cloud
- For big objects, Swift is preferred to Amazon as it is cheaper



50 mixed sequences, Mutli-Cloud experiment. Used storage per Cloud

Evaluation – STaaS requests SLA Violations

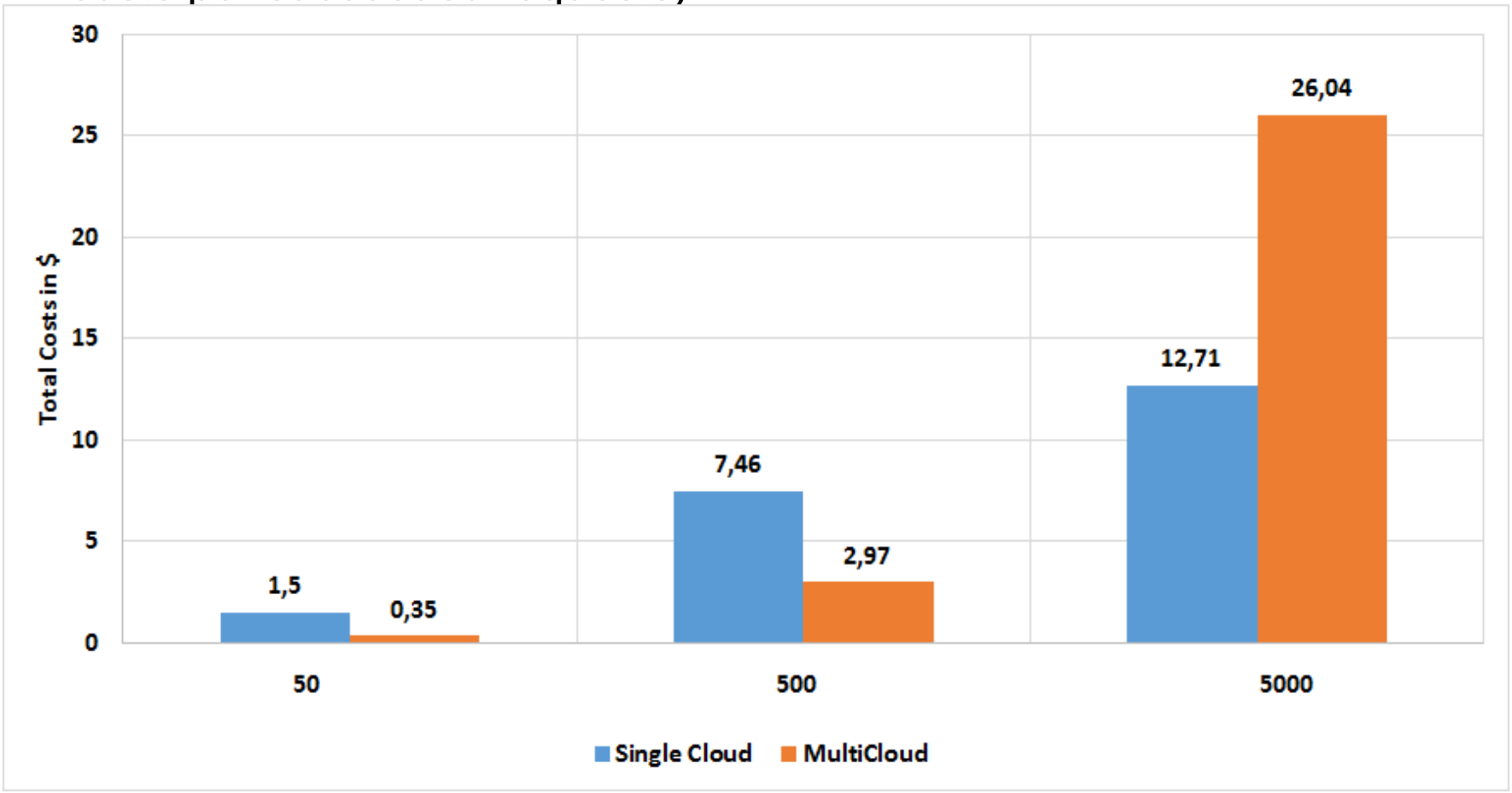
- Due to over capacity, all sequences after minute 28000 are declined.
- Only 15 requests failed due to SLA violation (object size can not be satisfied)



Single Cloud, 5000 mixed sequence types

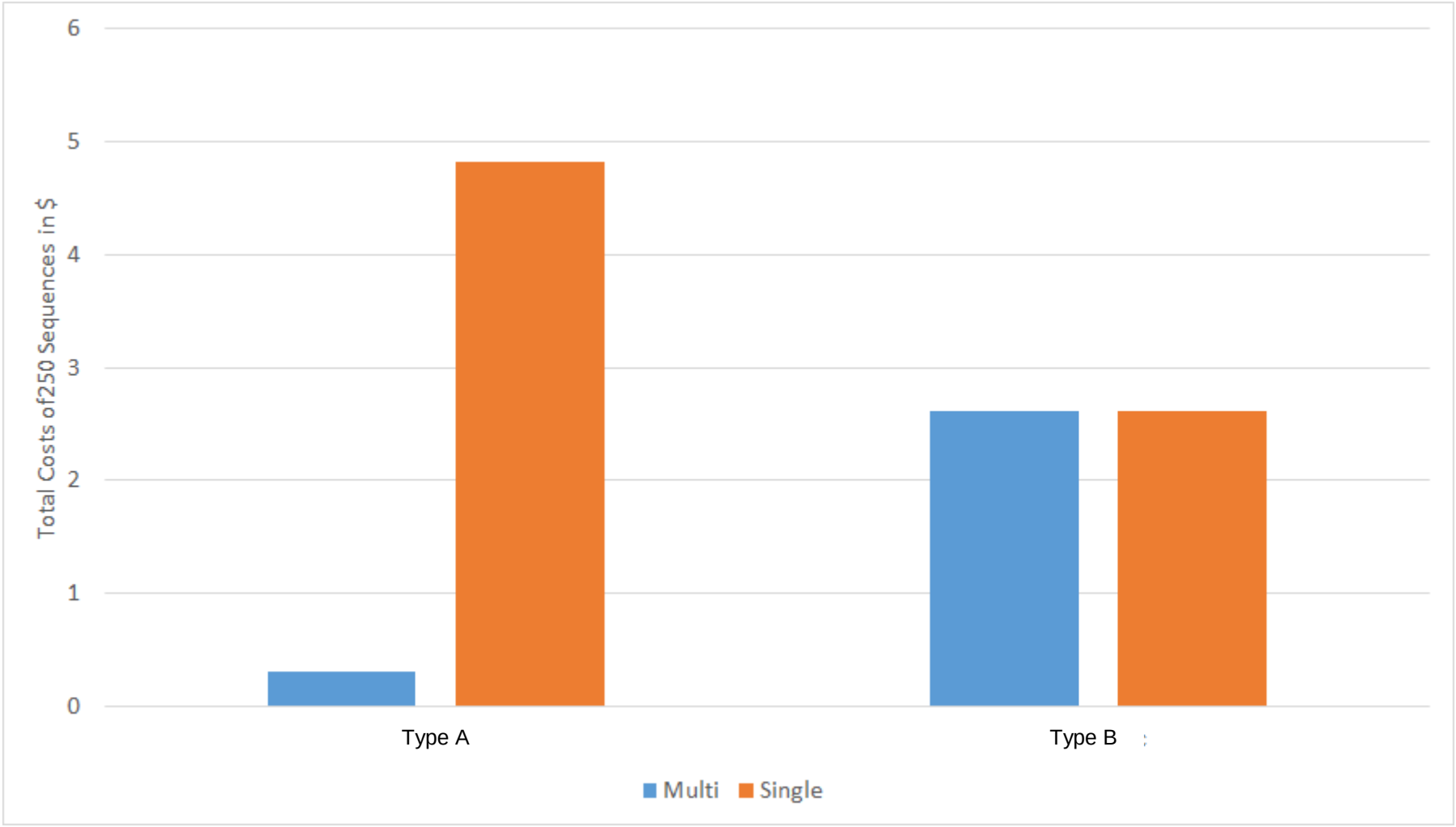
Evaluation – Total Usage Costs

- Multi-Cloud is more cost-saving compared to single Cloud (in terms of costs per succeeded requests)



50, 500 and 5000 mixed sequences – Mutli & Single Cloud

Evaluation – Effect of `UsageSquence` Type



Conclusion

- Development of STaaS extension for *CloudSim* simulation environment
- Modelling of different SLA requirements and SLA matching policies for STaaS Clouds
- Evaluation of used Storage and costs for single and multi-Cloud scenario with different `UsageSquence` types
- Usage of multiple Clouds lead to cost-savings if SLA is not too restrictive

Future Work

- More complex SLA matching policies (location, throughput)
- More complex price models for STaaS
- Dynamic Broker Decisions
- Modeling of different storage controller policies (example: *OpenStack Swift Ring*)

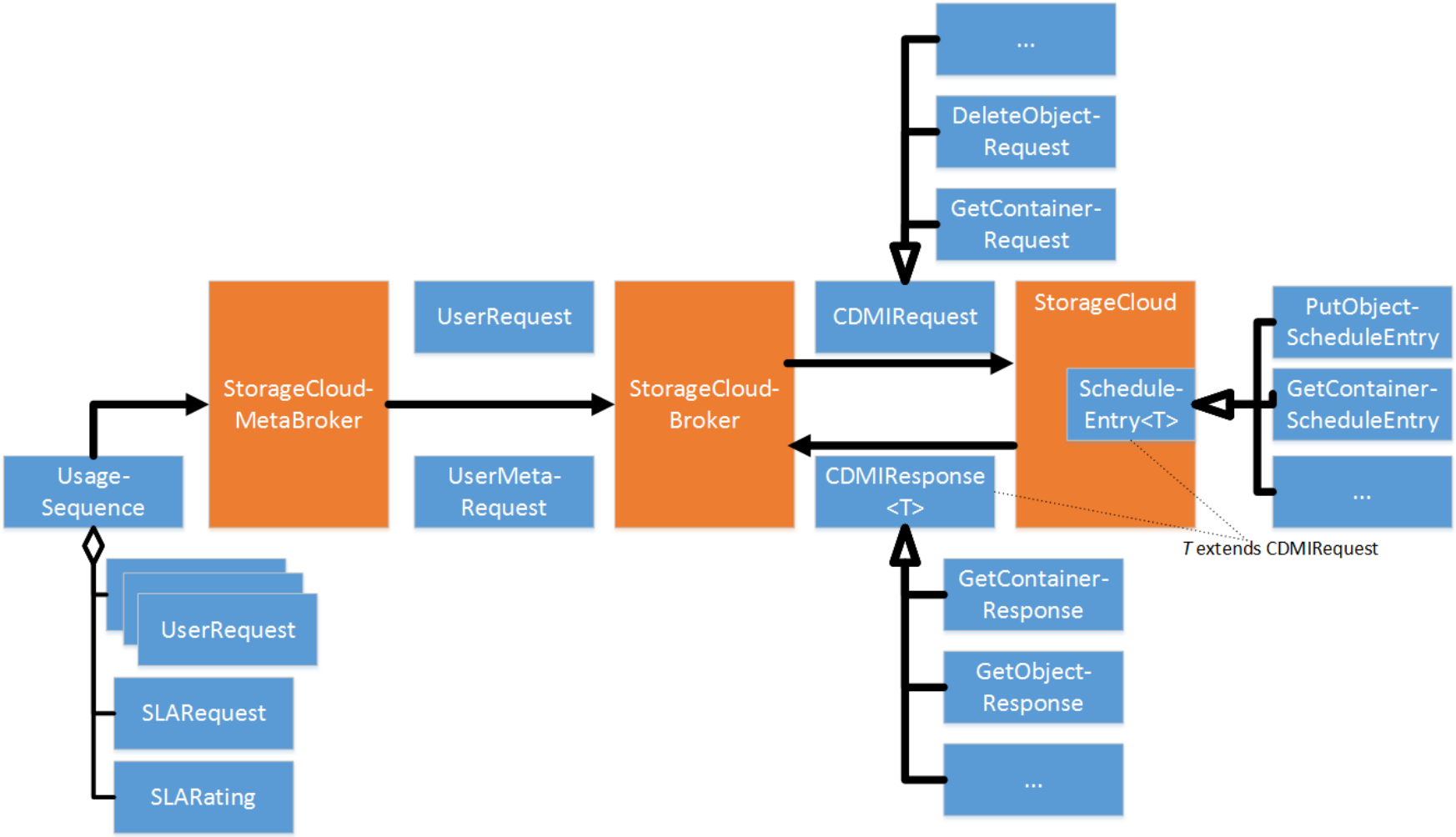
References

- [1] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar AF De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1):23–50, 2011.
- [2] Rodrigo N Calheiros, Rajiv Ranjan, C´esar AF De Rose, and Rajkumar Buyya. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. arXiv preprint arXiv:0903.2525, 2009.
- [3] Cloud Data Management Interface (CDMI) Version 1.0.2.
- [4] Amazon Inc. Amazon s3, cloud computing storage for files, images, videos, 2013. [Online; accessed 17-July-2013].
- [5] Peter Mell and Timothy Grance. The nist definition of cloud computing (draft). NIST special publication, 800(145):7, 2011.
- [6] A. Schill and T. Springer. Verteilte Systeme: Grundlagen und Basistechnologien. Springer London, Limited, 2007.

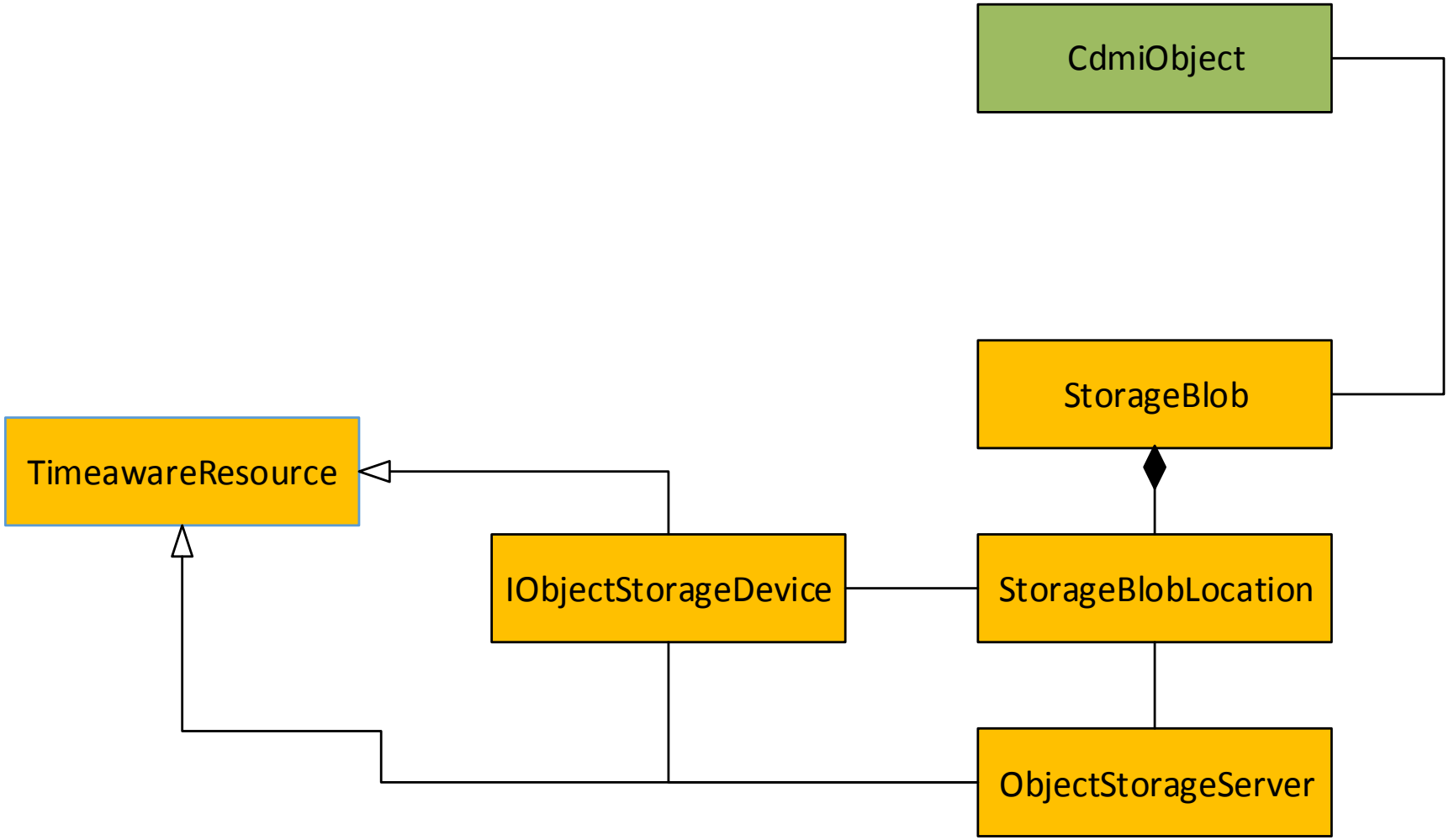
Karlsruher Institut für Technologie

Questions ?

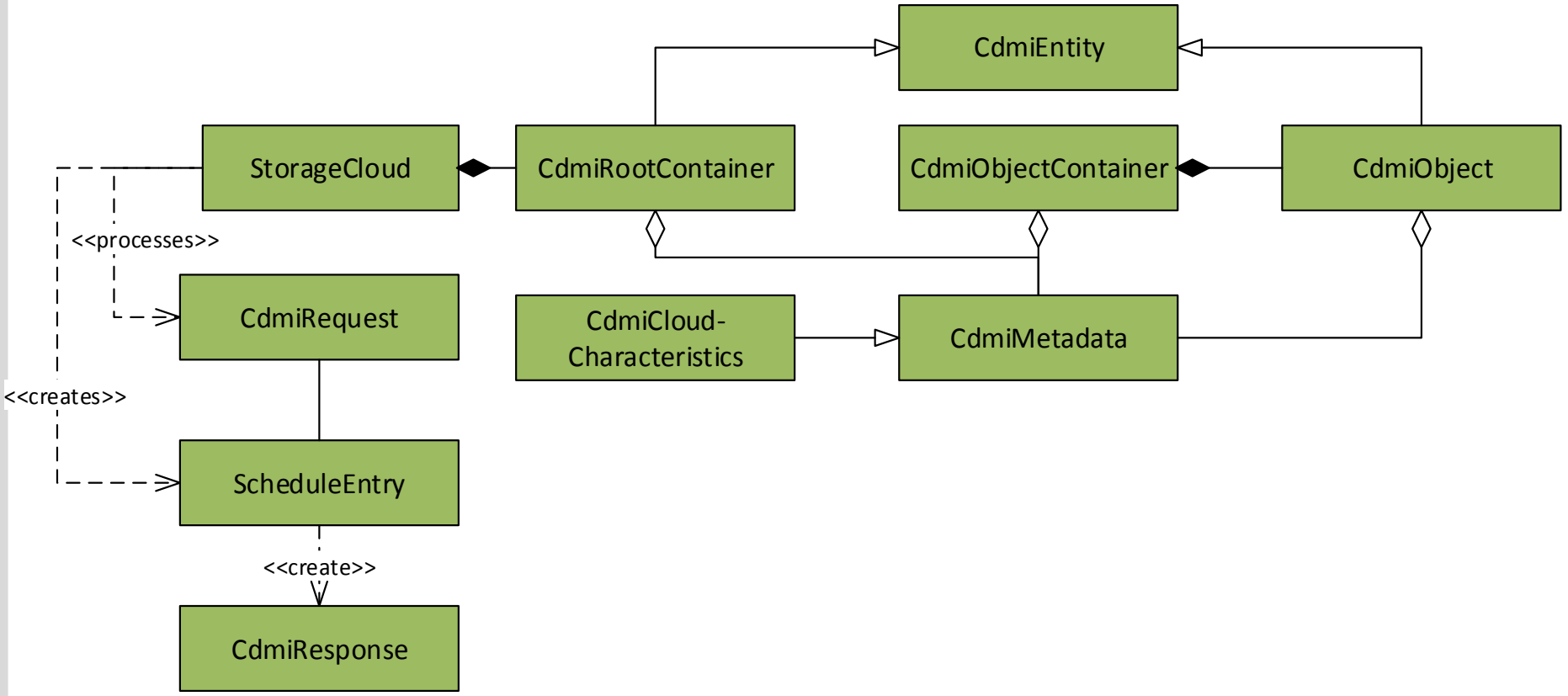
Implementation – Cloud User Interface



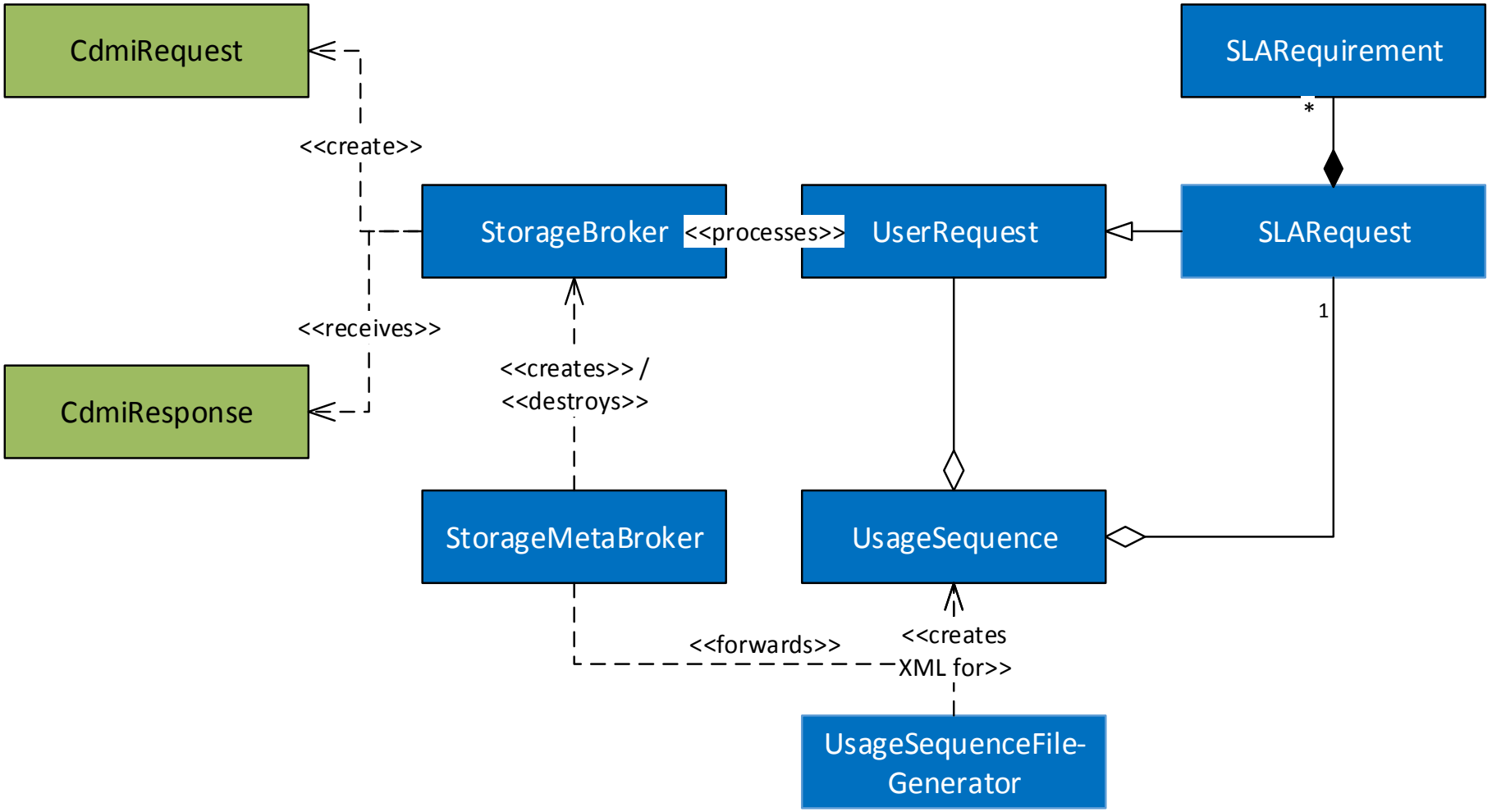
Implementation – Storage Models



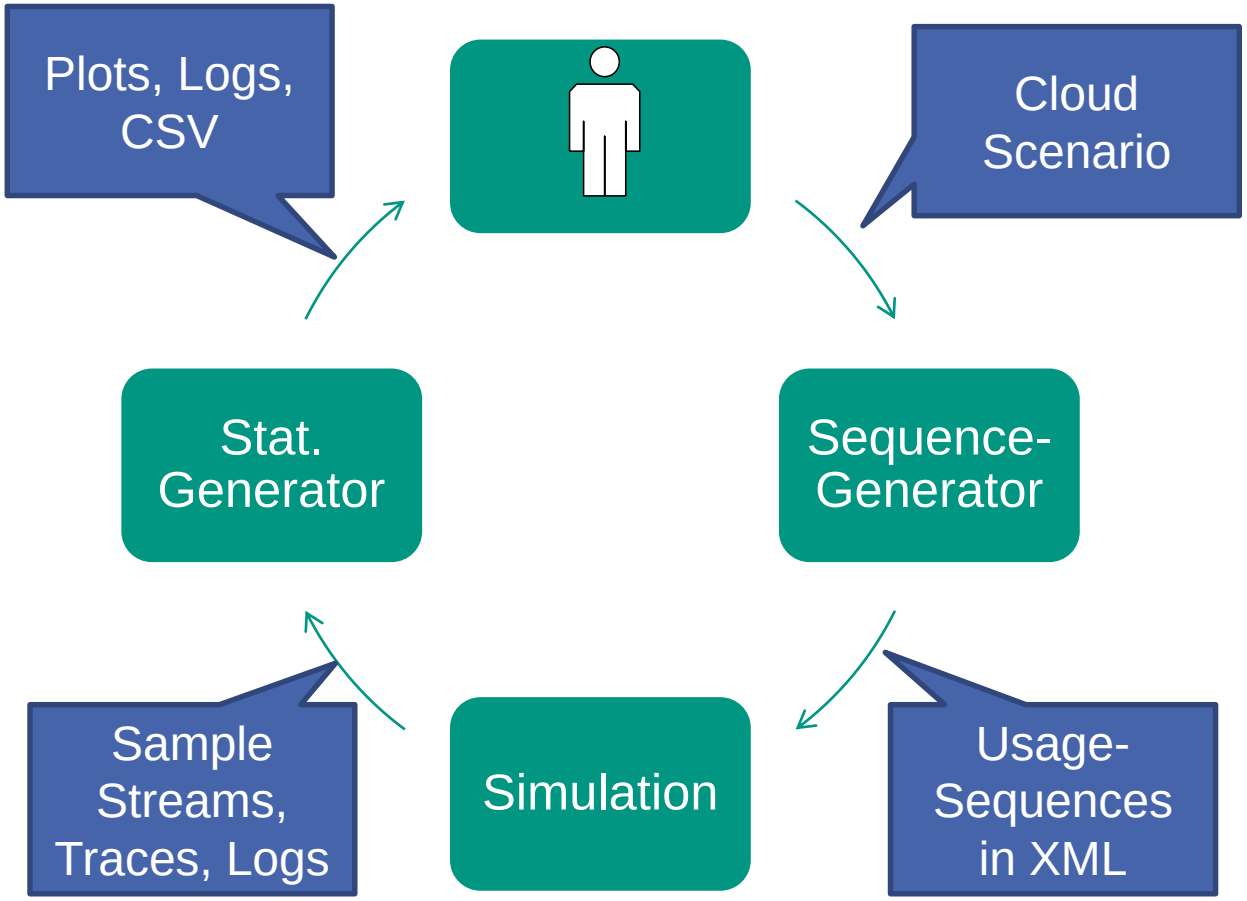
Implementation – Storage Models



Implementation – Usage Models



Simulation Workflow



Implementation – Provider Side Hardware Modeling