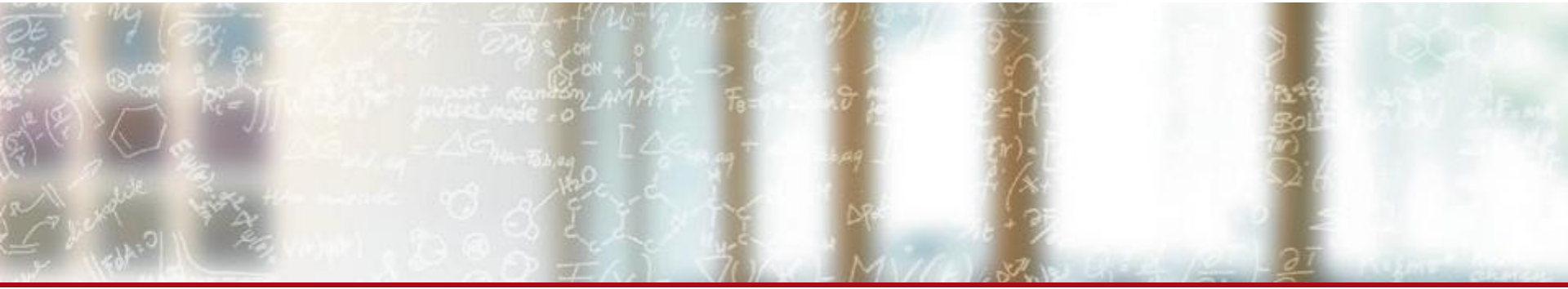
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



FirecREST: RESTful API on Cray XC systems

Felipe Cruz
Swiss National Supercomputing Centre, ETH Zurich, Lugano, Switzerland
felipe.cruz@cscs.ch

FirecREST, what is it and why it matters?

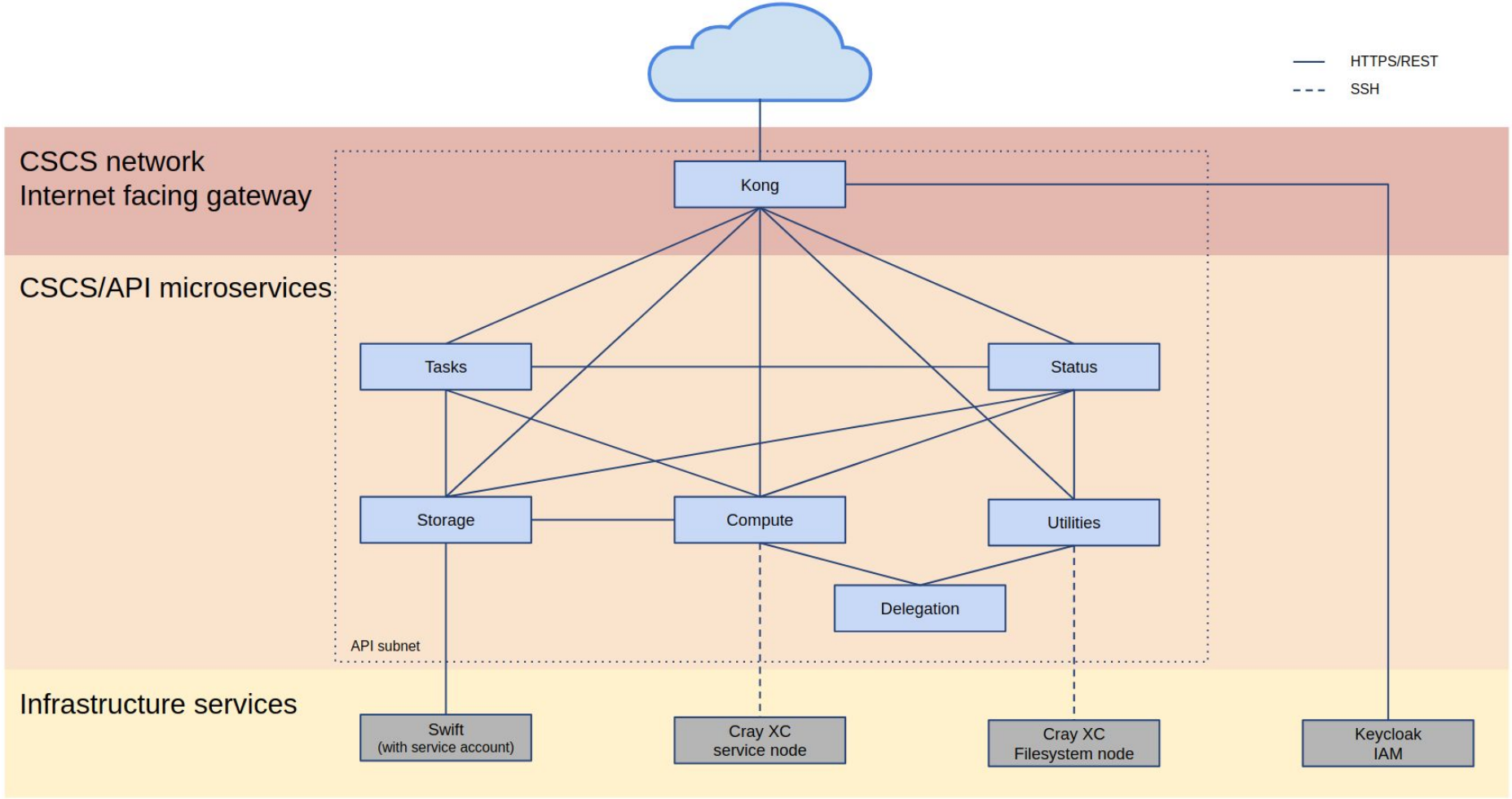
- What is FirecREST? a RESTful Services Gateway to HPC resources based on a microservices architecture.
  - RESTful interfaces. Architecture abstraction that allows for applications to be simple, lightweight, and fast.
  - Microservice architecture. Decouple functionalities in traditional monolithic servers architecture.
  - HPC resources. Integrates with infrastructure available at HPC centers:
    - Identity Access Management
    - Workload management
    - Data mover

FirecREST, what is it and why it matters?

- Why FirecREST? Allows web applications to securely access user-owned HPC resources
  - enforce that all API requests are authenticated
  - applications never manipulate user credentials
  - only allow requests from registered applications
  - user-managed access permissions per application
  - stateless security model by use of tokens
  - enables managing execution of workloads
  - enables external transfer of data from/to HPC filesystems

FirecREST motivation / use-cases

FirecREST architecture


HTTPS/REST
SSH
CSCS network
Internet facing gateway
CSCS/API microservices
Infrastructure services
Kong
Tasks
Status
Storage
Compute
Utilities
Delegation
API subnet
Swift
(with service account)
Cray XC
service node
Cray XC
Filesystem node
Keycloak
IAM

FirecREST architecture components

Services

- Kong API gateway
- Compute
- Storage
- Utilities
- Support
  - Tasks
  - Status
  - Delegation

CSCS core services dependencies

- Workload Manager node
  - Compute
  - Xfer
- Utility node, filesystem utilities
- SWIFT, data transfer staging area
- Keycloak, authentication and authorization

FirecREST microservices

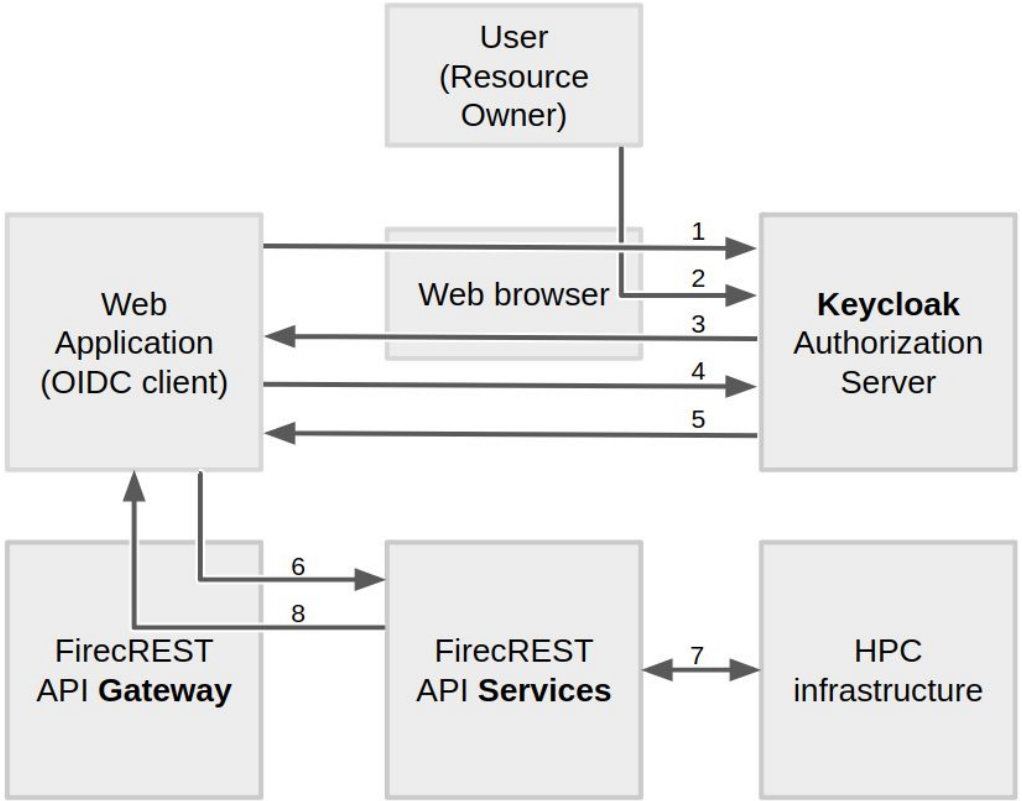
- **Kong**: Open-Source microservice API Gateway. Implements and **enforces Authentication, Authorization**, Traffic control, Analytics, Logging.
- **Compute**: Non-blocking calls to workload manager to submit and query jobs. The service responds with a reference to the temporary task resource that tracks the state of the request.
- **Storage**: Non-blocking calls to high-performance storage services. The service responds with a reference to the temporary task resource that tracks the state of the request.

FirecREST microservices

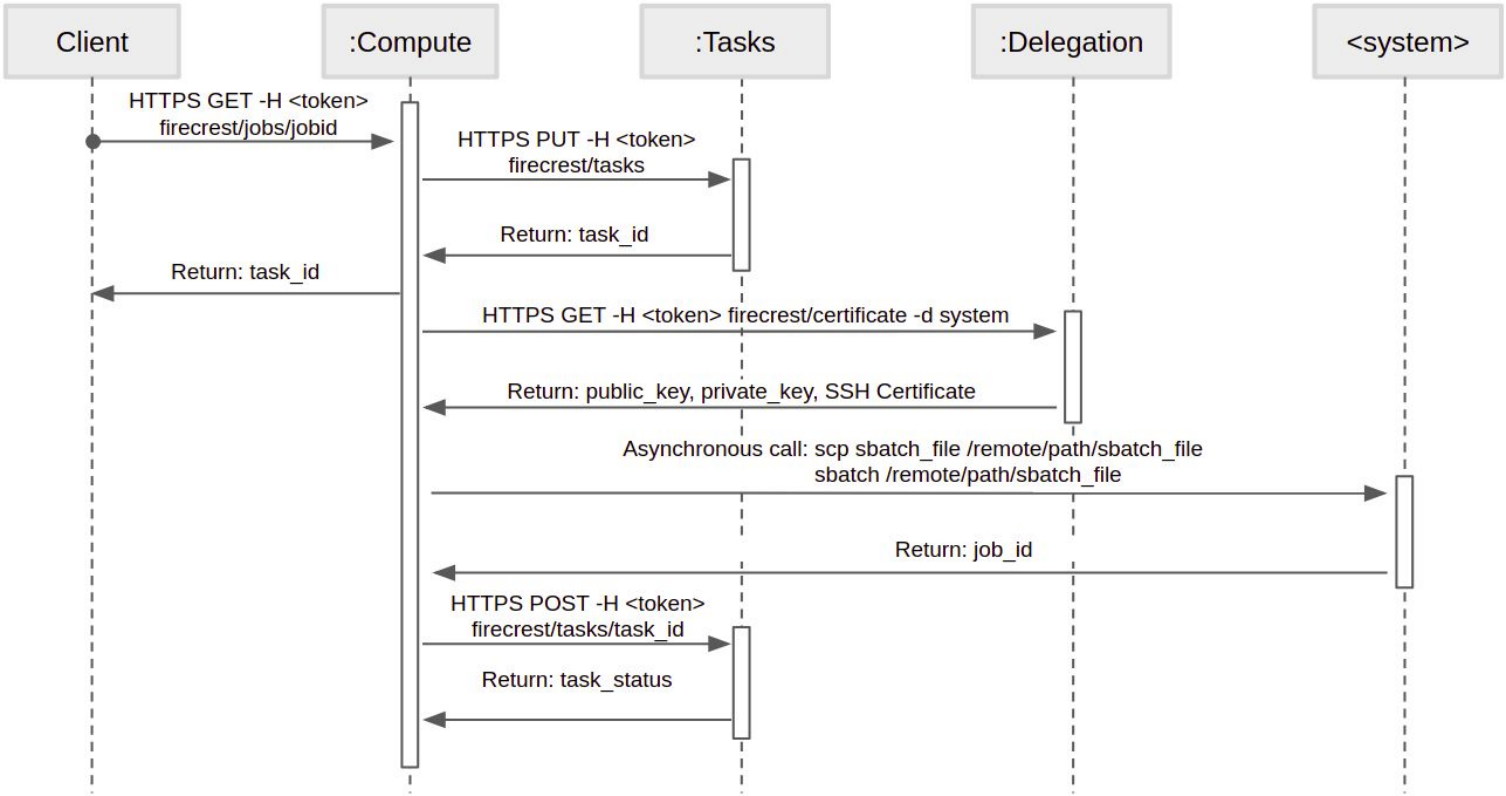
- **Utilities**: Fundamental filesystem utilities. All calls are blocking operations, maximum operation duration is limited by a timeout.
- **Support**
  - *Tasks*: state of tasks. Tracking of non-blocking compute and storage operations.
  - *Delegation*: Delegation with conversion from OIDC token to SSH user-certificate.
  - *Status*: information on services and infrastructure.

FirecREST core workflows

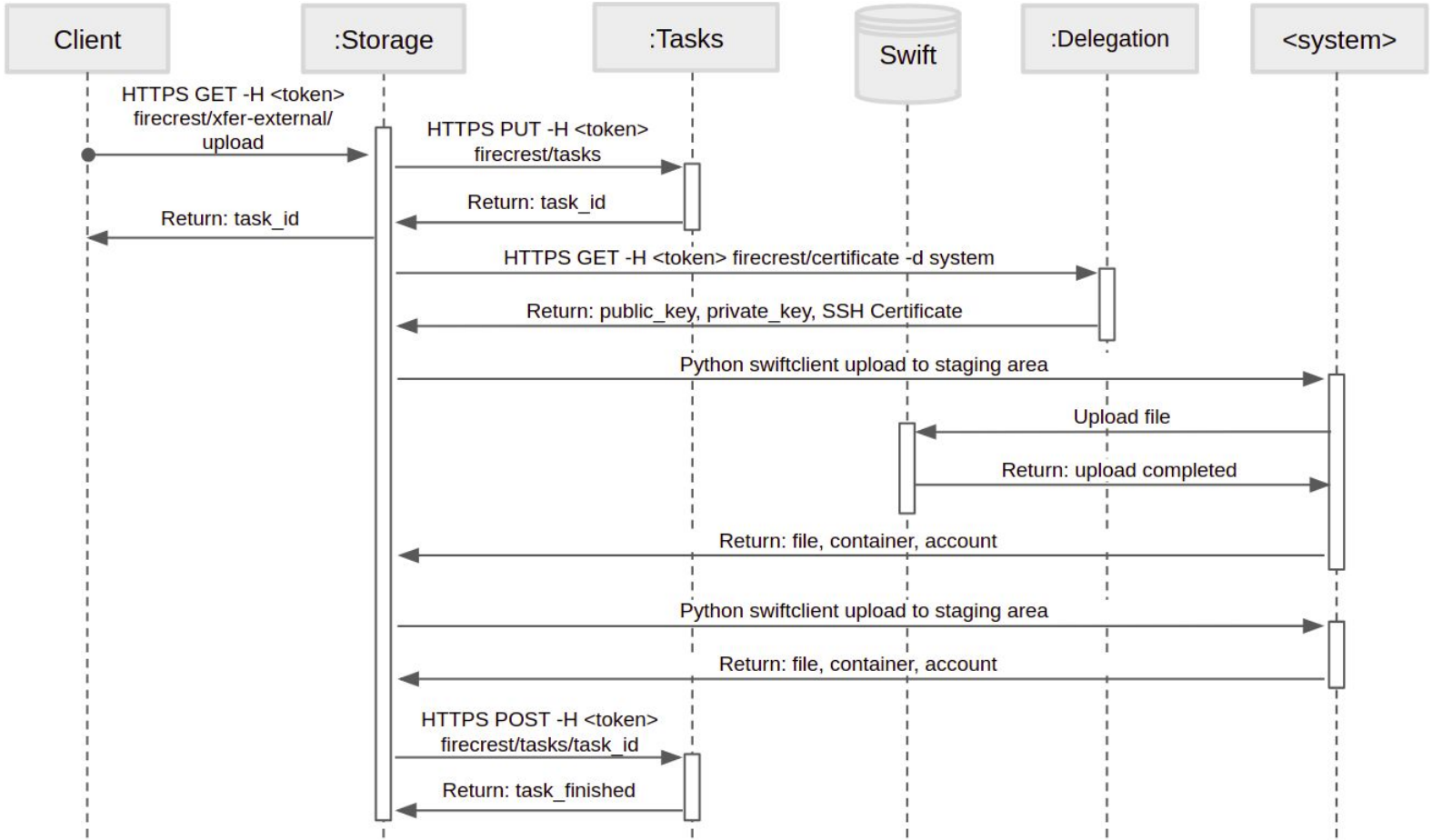
FirecREST workflow: Authentication



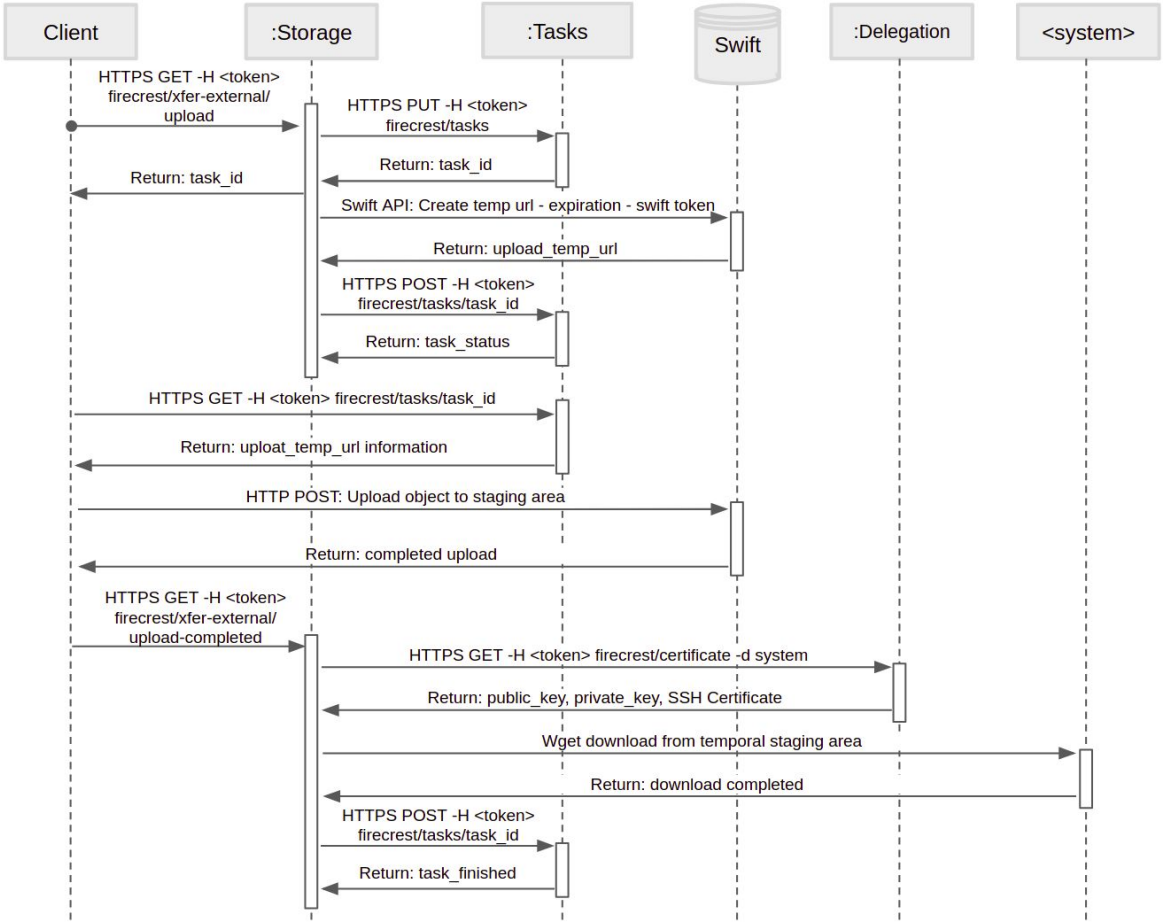
FirecREST workflow: Job submission



FirecREST workflow: Download data



FirecREST workflow: Upload data



FirecREST API overview

Compute

Non-blocking calls to workload manager to submit and query jobs. The service responds with a reference to the temporary task resource that tracks the state of the request.

POST	/Jobs/{machine}	Submit Job
GET	/Jobs/{machine}	Retrieves information from all jobs
GET	/Jobs/{machine}/{jobid}	Retrieves information from a job
DELETE	/Jobs/{machine}/{jobid}	Delete Job
GET	/Jobs/{machine}/sacct	Job account information

Storage

Non-blocking calls to high-performance storage services. The service responds with a reference to the temporary task resource that tracks the state of the request.

GET	`/Storage/xfer-internal/rsync`	rsync
GET	`/Storage/xfer-internal/mv`	move (rename) files
GET	`/Storage/xfer-internal/cp`	copy files and directories
GET	`/Storage/xfer-internal/rm`	remove files or directories
GET	`/Storage/xfer-external/upload`	Upload a file
GET	`/Storage/xfer-external/download`	Download a file

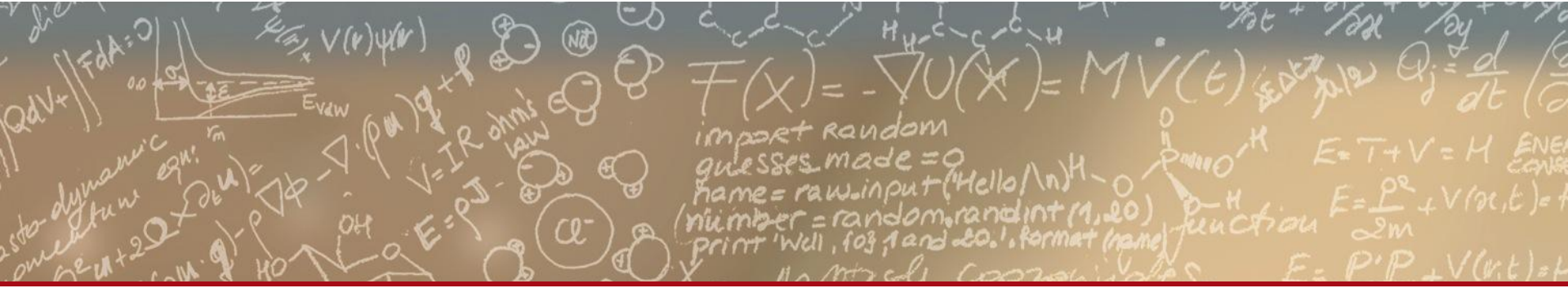
Queue Access status and response of compute and storage tasks.

GET	/Queue/tasks	returns all tasks
PUT	/Queue/tasks	Creates a task
GET	/Queue/tasks/{id}	task status information
POST	/Queue/tasks/{id}	Updates a task
DELETE	/Queue/tasks/{id}	Delete task

Utilities

Basic system utilities. All calls are blocking and low-latency operations, maximum operation duration is limited by a timeout.

GET	`/utilities/{machinename}/ls`	List directory contents
POST	`/utilities/{machinename}/mkdir`	Creates a directory
POST	`/utilities/{machinename}/rename`	Rename/move a file, directory, or symlink
POST	`/utilities/{machinename}/chmod`	Change file mode bits
POST	`/utilities/{machinename}/chown`	Change file owner and group
POST	`/utilities/{machinename}/file`	determine file type
POST	`/utilities/{machinename}/symlink`	Create a symlink


import random
guesses_made = 0
name = raw_input('Hello! What is your name?\n')
number = random.randint(1, 20)
print 'Well, ' + name + ', I am thinking of a number between 1 and 20.'

Thank you