

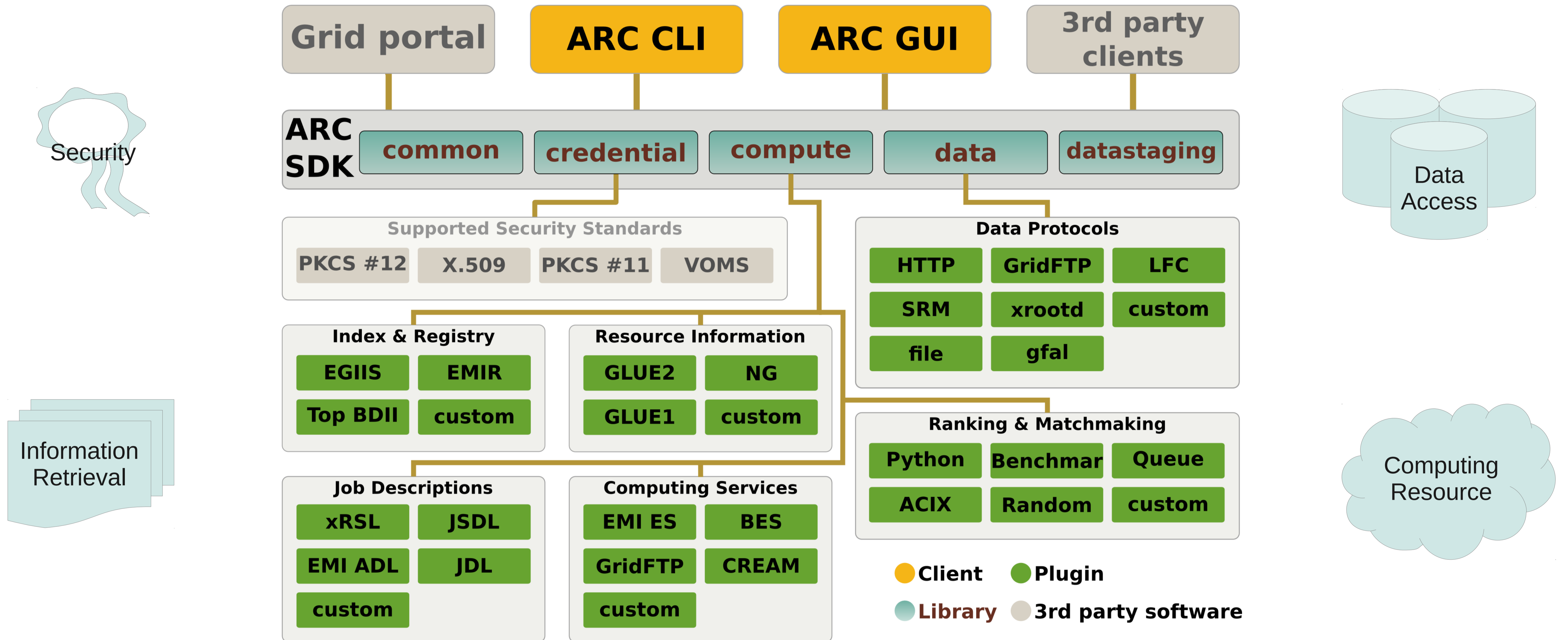
ARC SDK

A toolbox for distributed computing and data applications

The simple way for your app to use the Grid



David Cameron (University of Oslo/NeIC, Norway)
Jonas Lindemann (Lund University/SNIC/Lunarc, Sweden)
Martin Skou Andersen (University of Copenhagen, Denmark)



Use the SDK to submit a Grid job in...

Python

```
import arc
import sys

# UserConfig contains information on credentials and default services to use.
usercfg = arc.UserConfig("", "")

# Simple job description which outputs hostname to stdout
jobdescstring = "%(executable=bin/hostname)(stdout=stdout)"

# Parse job description
jobdescs = arc.JobDescriptionList()
if not arc.JobDescription_Parse(jobdescstring, jobdescs):
    print "Invalid job description"
    sys.exit(1)

# Use top-level NorduGrid information index to find resources
index = arc.Endpoint("ldap://index1.nordugrid.org:2135/Mds-Vo-name=NorduGrid.o=grid",
                    arc.Endpoint.REGISTRY,
                    "org.nordugrid.idapegis")
services = arc.EndpointList(1, index)

# Do the submission
jobs = arc.JobList()
submitter = arc.Submitter(usercfg)
if submitter.BrokeredSubmit(services, jobdescs, jobs) != arc.SubmissionStatus.NONE:
    print "Failed to submit job"
    sys.exit(1)

# Write information on submitted job to local job list (-./arc/jobs.xml)
jobList = arc.JobInformationStorageXML(usercfg.JobListFile())
if not jobList.Write(jobs):
    print "Failed to write to local job list %s" % usercfg.JobListFile()

# Job submitted ok
print "Job submitted with job id %s" % jobs.front().JobID
```

C++

```
#include <arc/UserConfig.h>
#include <arc/compute/Endpoint.h>
#include <arc/compute/Job.h>
#include <arc/compute/JobDescription.h>
#include <arc/compute/Submitter.h>
#include <arc/compute/JobInformationStorageXML.h>

int main() {
    // UserConfig contains information on credentials and default services to use.
    Arc::UserConfig usercfg("", "");

    // Simple job description which outputs hostname to stdout
    std::string jobdesc("%(executable=bin/hostname)(stdout=stdout)");

    // Parse job description
    std::list<Arc::JobDescription> jobdescs;
    if (!Arc::JobDescription_Parse(jobdesc, jobdescs)) {
        std::cerr << "Invalid job description" << std::endl;
        return 1;
    }

    // Use top-level NorduGrid information index to find resources
    Arc::Endpoint index("ldap://index1.nordugrid.org:2135/Mds-Vo-name=NorduGrid.o=grid",
                      Arc::Endpoint.REGISTRY,
                      "org.nordugrid.idapegis");
    std::list<Arc::Endpoint> services(1, index);

    // Do the submission
    std::list<Arc::Job> jobs;
    Arc::Submitter submitter(usercfg);
    if (submitter.BrokeredSubmit(services, jobdescs, jobs) != Arc::SubmissionStatus::NONE) {
        std::cerr << "Failed to submit job" << std::endl;
        return 1;
    }

    // Write information on submitted job to local job list (-./arc/jobs.xml)
    Arc::JobInformationStorageXML jobList(usercfg.JobListFile());
    if (!jobList.Write(jobs)) {
        std::cerr << "Failed to write to local job list" << usercfg.JobListFile() << std::endl;
    }

    // Job submitted ok
    std::cout << "Job submitted with job id " << jobs.front().JobID << std::endl;
    return 0;
}
```

Java

```
import nordugrid.arc.*;

public class BasicJobSubmission {
    public static void main(String argv[]) {
        // UserConfig contains information on credentials and default services to use.
        UserConfig usercfg = new UserConfig("", "");

        // Simple job description which outputs hostname to stdout
        String jobdesc = "%(executable=bin/hostname)(stdout=stdout)";

        // Parse job description
        JobDescriptionList jobdescs = new JobDescriptionList();
        if (!JobDescription_Parse(jobdesc, jobdescs).toBool()) {
            System.err.println("Invalid job description");
            System.exit(1);
        }

        // Use top-level NorduGrid information index to find resources
        Endpoint index = new Endpoint("ldap://index1.nordugrid.org:2135/Mds-Vo-name=NorduGrid.o=grid",
                                     Endpoint.CapabilityEnum.REGISTRY,
                                     "org.nordugrid.idapegis");
        EndpointList services = new EndpointList();
        services.add(index);

        // Do the submission
        JobList jobs = new JobList();
        Submitter submitter = new Submitter(usercfg);
        if (!submitter.BrokeredSubmit(services, jobdescs, jobs)
            .equals(nordugrid.arc.SubmissionStatus.SubmissionStatusType.NONE)) {
            System.err.println("Failed to submit job");
            System.exit(1);
        }

        // Write information on submitted job to local job list (-./arc/jobs.xml)
        JobInformationStorageXML jobList = new JobInformationStorageXML(usercfg.JobListFile());
        if (!jobList.Write(jobs)) {
            System.err.println("Failed to write to local job list " + usercfg.JobListFile());
        }

        // Job submitted ok
        System.out.println("Job submitted with job id " + jobs.begin().next().getJobID());
        return;
    }
}
```

ARC Storage Explorer: A Grid data management GUI using the ARC Python SDK

