



GAT Status Review Meeting, March 2004

Tom Goodale

goodale@cct.lsu.edu

Albert-Einstein-Institut

and

LSU Center for
Computation and
Technology

GAT: Grid Application Toolkit

- API and Toolkit for developing portable Grid applications independently of the underlying Grid infrastructure and available services
 - Implements the GAT-API
 - Used by applications (different languages)
- GAT Adaptors
 - Connect to capabilities/services
- GAT Engine
 - Provides the function bindings for the GAT-API

Application

“ Is there a better resource I could be using?”

SOAP

WSDL

CORBA

OGSA

Other

Monitoring

Security

Profiling

Information

Logging

Notification

*Data
Management*

*Resource
Management*

*Application
Manager*

Migration

GLOBUS

UNICORE

*Other Grid
Infrastructure?*

The Same Application ...

Laptop

Super Computer

The Grid

Application

Application

Application

GAT

GAT

GAT



No network!



Firewall issues!



Application

“Is there a better resource I could be using?”

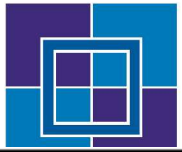
newmachine = GAT_Resource.Find()



The Grid

- Application makes GAT API calls for operations which may be Grid-related.
- Application links against the GAT Engine
- Application runs irrespective of actual underlying infrastructure deployment
 - Engine loads adaptors which are valid in the environment extant when the application starts
 - Adaptors try to do Grid operations on request, on failure another adaptor provided function may be called.
- Application can thus be compiled, linked and tested without any Grid services
- Same application executable can run in a full Grid environment

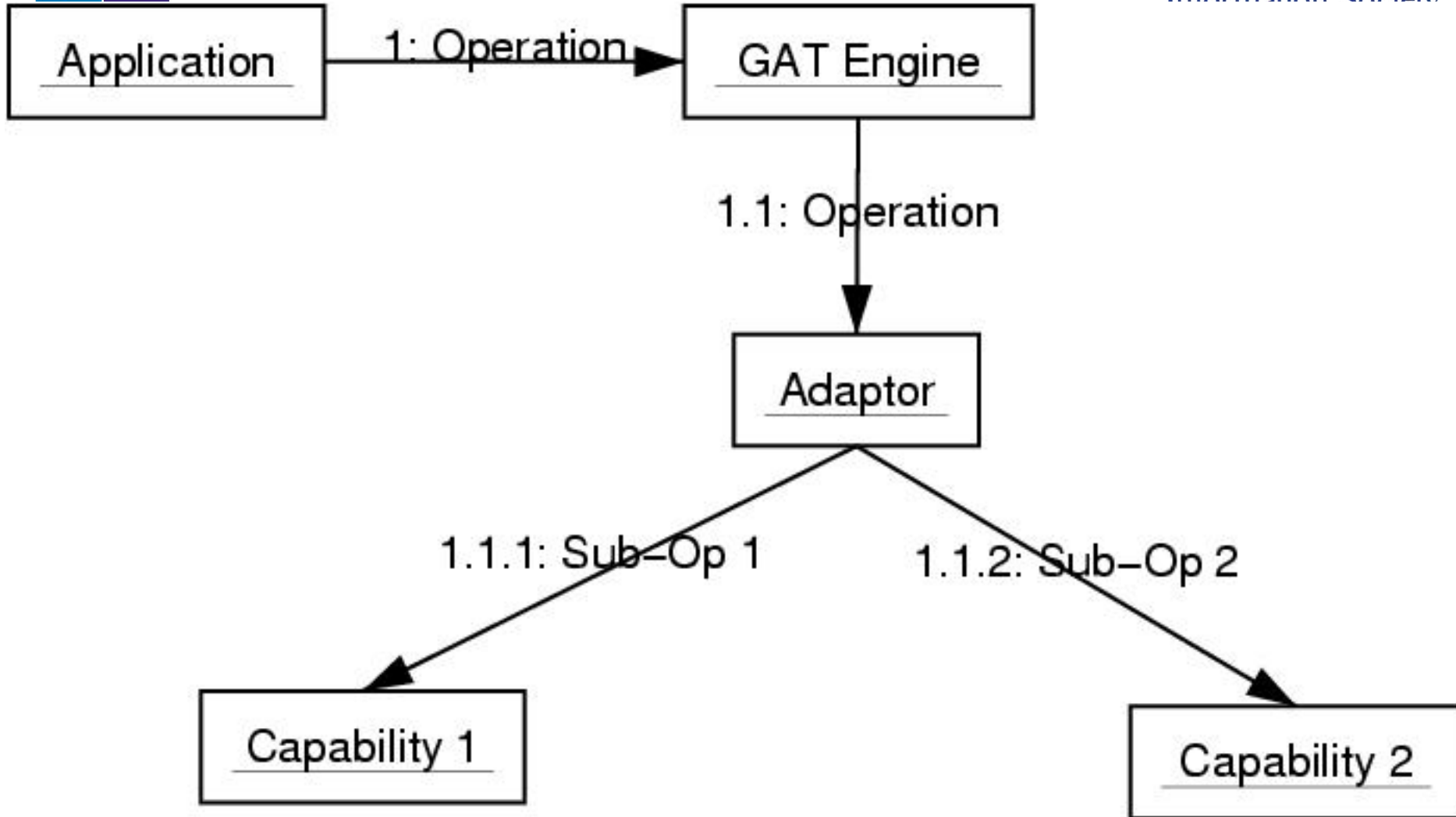
- The GAT uses whatever underlying Grid infrastructure there is and that people have developed adaptors for.
- GAT is not about replacing already developed infrastructure, but instead to provide a simple, clear interface which can be used with many different infrastructures.
 - Different versions of Globus
 - Condor
 - Unicore
 - ...



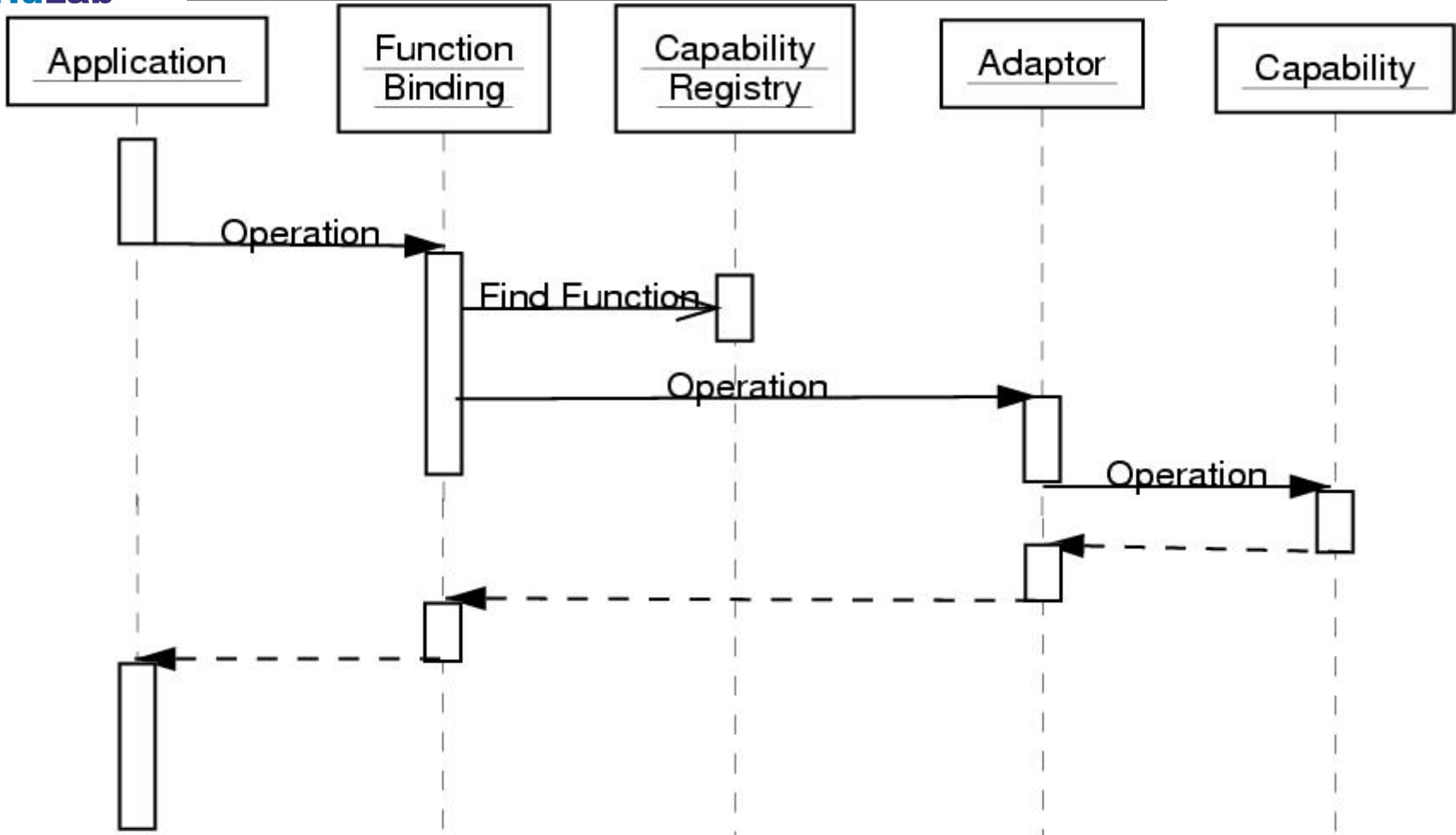
Engine Calling Adaptor



Information Society



An API Call



- The GAT must support applications written in any language which people write Grid Applications in:
 - C, C++, Fortran, Java, Perl, Python, ...
- The use of the GAT API should be as natural as possible for users of these languages.
- It must also not require a steep learning curve to move from the API in one language to the API in another language
 - APIs in different languages should be as similar as possible

- At the last review we had a prototype implementation and had started working on the full API specification.
- Now have the full API specification, and a C reference version of the GAT Engine and adaptors which implement this specification.
 - This will be used today and at GGF to demonstrate the GridLab migration scenario.

- Simple adaptors available for all but one of the GAT capabilities.
- GridLab adaptors being developed for all capabilities
 - Currently have adaptors for Data and some GRMS functionality.
- Unicore people working on an adaptor.
- Work has started on a C++ wrapper.

- Review implementation and resolve issues which arose with API specification.
 - Re-issue and re-review API-specification
- Document C and C++ bindings.
- Finish users and adaptor-developer guides.
- Develop Fortran, Perl, Python and Java APIs
 - Will use SWIG to provide bindings to C reference implementation.
- Work with Triana and Portal packages to provide a native Java implementation which may be tested against the wrapped reference implementation.

- Work with service developers to provide adaptors.
- Use GAT to Grid-enable as large a class of applications as possible
 - already in discussions with lots of application groups.
- Review:
 - There are bound to be problems with provided functionality and the usability of the specification, and there will be feature requests.

- Release version 2 of API specification in July.
- Revise the language specific API specifications in line with this new version.
- Update all implementations to the new version
 - C reference implementation
 - Adaptors
 - Other language wrappers
 - Java native implementation
- Update user code to use new revision.

- Have proposed a GGF working group to develop this API into a standard.
 - First BoF at GGF10 in Chicago was very successful
- Working group should bring in many more user groups and service developers
 - After BoF had expressions of interest from industry (e.g. Cisco and Intel) and from many application groups.
- Apps-TWG of the GridStart project will be examining the GAT from the application perspective.