

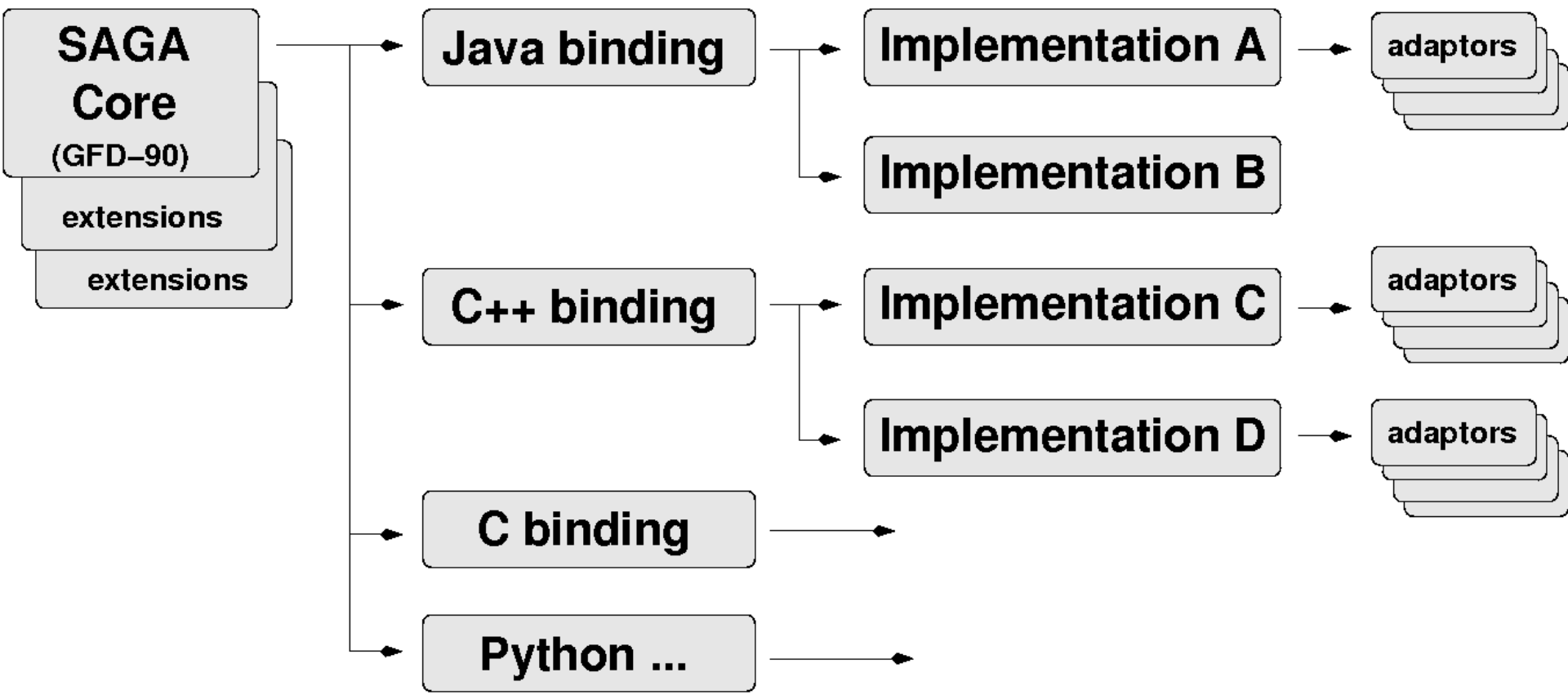
SAGA for Java Applications

Thilo Kielmann

Vrije Universiteit, Amsterdam



The SAGA Landscape



SAGA for Java Applications

- Java language binding
- Implementation
- Time lines

SAGA Language Bindings

Language bindings, according to GFD.90:

- Provide syntax
- Should maintain the language's “Look and Feel”
- Resolve all details left open in GFD.90
 - Thread safety, concurrency
 - Object lifetime
 - Combination with related language features
 - ...
- The Java binding is our (SAGA-CORE-WG) first such exercise

A Java Language Binding

- Equivalent to “C-like header files”:
 - SAGA describes classes and interfaces
 - Java-binding uses interfaces only (plus factory classes)
 - This allows that all implementations **MUST** use our binding interfaces
 - Gives 100% implementation compatibility

Using Factories (file copy)

```
public class FileTest {
    public static void main(String[] args) throws Exception {
        Session session = SessionFactory.createSession(true);
        Context context = ContextFactory.createContext();
        session.addContext(context);

        URL src = new URL(args[0]);
        URL dest = new URL(args[1]);

        File source = FileFactory.createFile(session, src);

        source.copy(dest, Flags.NONE.getValue());
        session.close(-1); // Shutdown this session.
    }
}
```

Java Binding: Operation Flags

- SAGA NameSpace and subclasses use various flags and OR them, in POSIX style
- Java has enumerations
 - Not equivalent to integers
 - Cannot be OR'ed directly
 - Cannot be subclassed (e.g., all File flags are in NameSpace)

org.ogf.saga.namespace.flags

- `public enum Flags extends Enum<Flags>`
- APPEND, CREATE, READWRITE, ...

Method Summary	
int	<code>getValue()</code> Returns the integer value of this enumeration literal.
boolean	<code>isSet(int val)</code> Tests for the presence of this flag in the specified value.
int	<code>or(Flags val)</code> Returns the result of or-ing this flag into another.
int	<code>or(int val)</code> Returns the result of or-ing this flag into an integer.
static <code>Flags</code>	<code>valueOf(String name)</code> Returns the enum constant of this type with the specified name.
static <code>Flags[]</code>	<code>values()</code> Returns an array containing the constants of this enum type, in the order they are declared.

Files

- SAGA files are POSIX style
- We use Java RandomFile
- InputStream and OutputStream to be layered on top
- No error codes, instead IOException

org.ogf.saga.file.file

- public interface File extends NSEntry
 - `int read(int len, Buffer buffer)`
 - `long seek(long offset, SeekMode whence)`
 - `int write(int len, Buffer buffer)`
 - ...

org.ogf.saga.task.task

- public interface Task extends SagaObject, Monitorable

Method Summary	
void	cancel() Cancels the asynchronous operation.
void	cancel(float timeoutInSeconds) Cancels the asynchronous operation.
SagaObject	getObject() Gets the object from which the task was created.
State	getState() Gets the state of the task.
void	rethrow() Throws any exception a failed task caught.
void	run() Starts the asynchronous operation.
void	waitTask() Waits for the task end up in a final state.
boolean	waitTask(float timeoutInSeconds) Waits for the task to end up in a final state.

More task interfaces

- **RVTask<E> extends Task**
- **Enum TaskMode extends Enum<TaskMode>**
 - ASYNC, SYNC, TASK
- Use as in (from File):

```
RVTask<Integer>  
read(TaskMode mode, int len,  
      Buffer buffer)
```

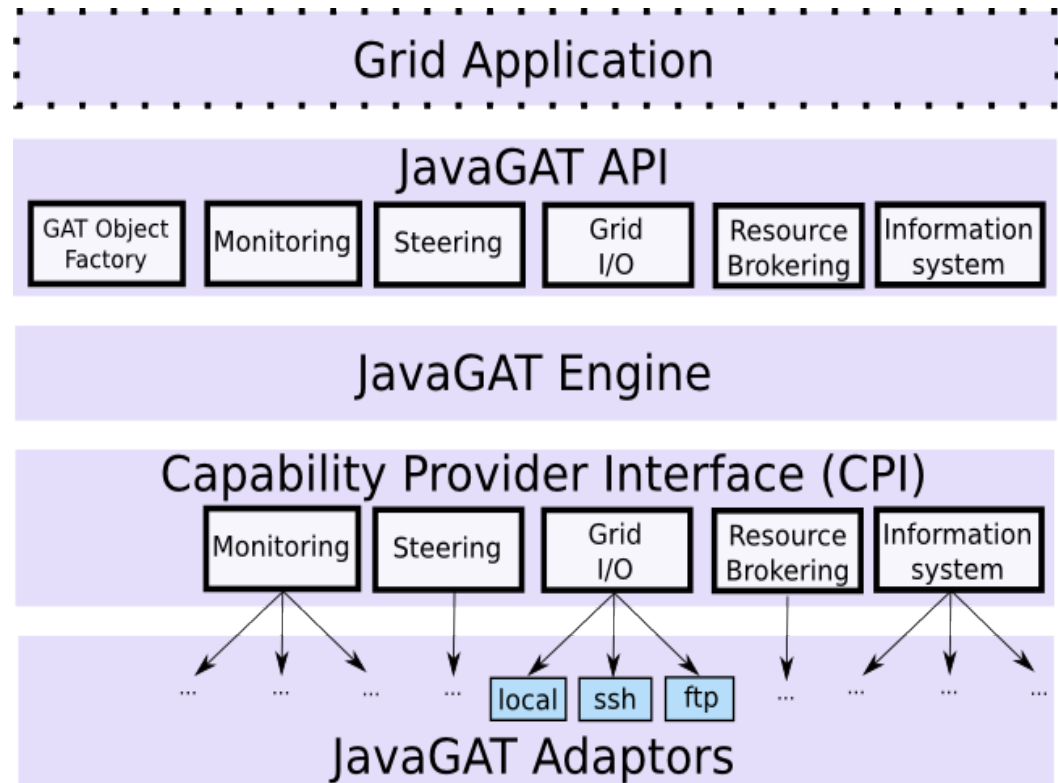
Implementing a Java-SAGA

- Pure Java solution needed:
“write once, run anywhere”
- Usefulness is equally determined by:
 - Language binding (look-and-feel)
 - Interfaces to middleware (applicability)
- We rely on our experience and code base from the JavaGAT, the Java Grid Application Toolkit

JavaGAT

- Earlier work, leading to SAGA
- Simple interface
- Lightweight engine
- Middleware adaptors:
 - local
 - Globus 2,3, (4)
 - SSH, SFTP
 - FTP, HTTP/S
 - PBS, SGE

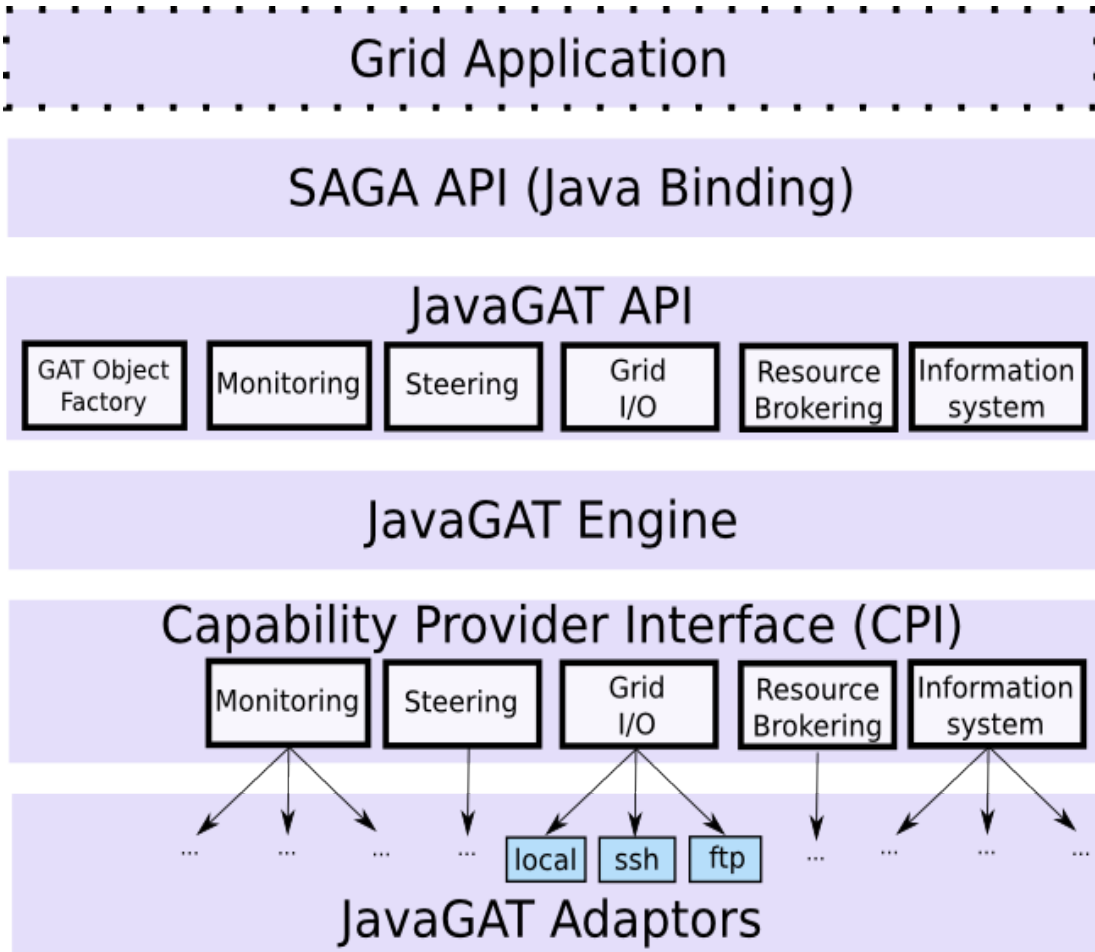
<http://www.cs.vu.nl/ibis/>



Strengths of JavaGAT

- Can cope with changing/imperfect middleware
- Intelligent dispatching of API calls to middleware
 - Dynamically finds out which adaptor works in a given environment, for a certain task
 - Caches adaptor selection
- Nested exceptions
 - Part of dynamic adaptor selection
 - Report to the use what is wrong in case of non-recoverable problems
- (see also paper in SC07)

Implementing Java SAGA



Timelines

- Java language bindings
 - To be discussed at OGF21(3:15, Leonesa I)
- Implementation
 - (incomplete) prototype SAGA-GAT engine to be demonstrated at SC07
 - Complete engine implementation 31/12/07
 - With all GAT adaptors (local, globus 2/3, ssh,...)
 - OMII-UK (GridSAM) adaptors 31/12/07
 - Globus 4 adaptors 31/03/08

Can't wait?

Existing SAGA Java implementations:

- EU-funded project DEISA has a partial implementation available
 - Files and Jobs, for the DEISA environment
- NAREGI project (Japan) has a prototype for its middleware stack