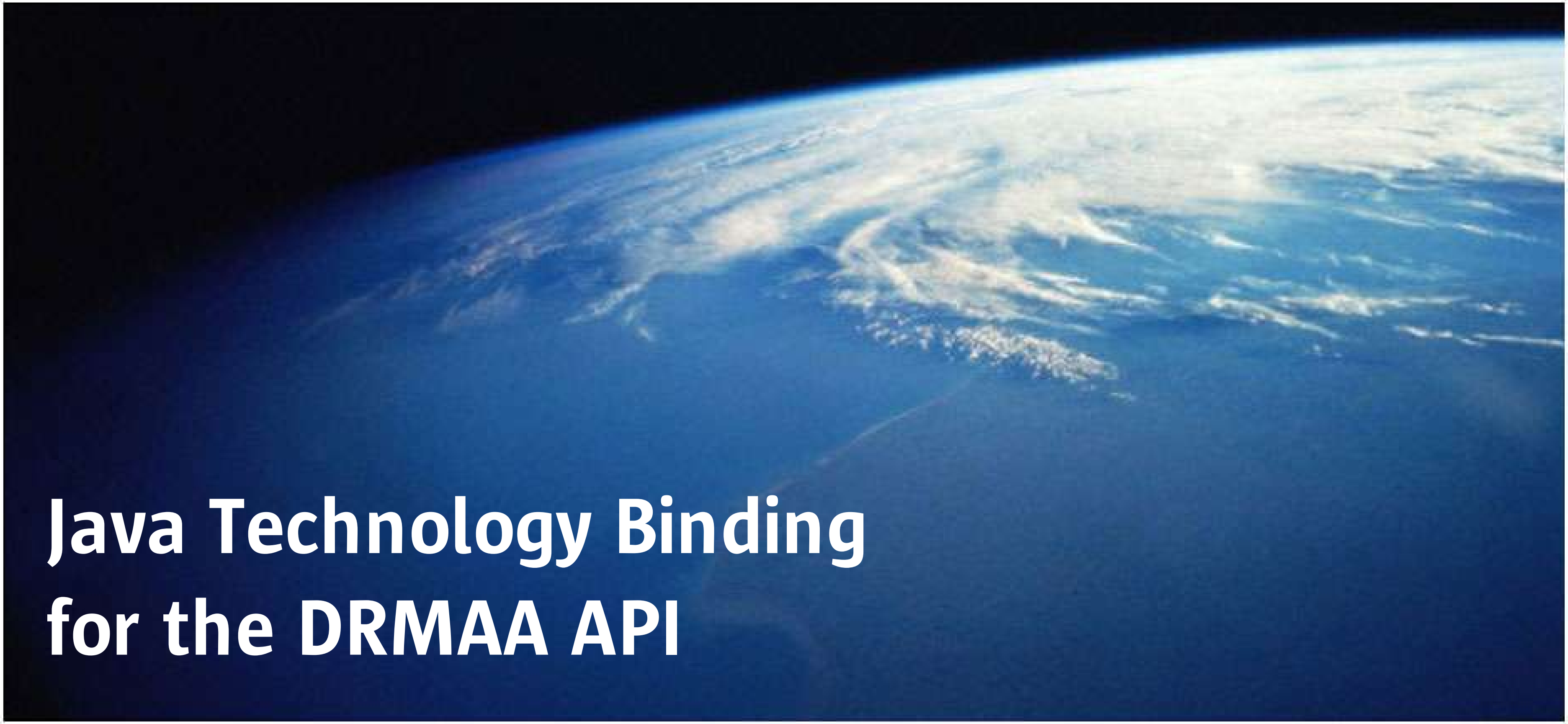

Java Technology Binding for the DRMAA API

Daniel Templeton

Staff Engineer

Sun Grid Engine

Sun microsystems

We make the net work.

Why Java™ Technology?

- Very popular language
  - Targeting 10 million developers
- Popular in grid computing
  - Well known
  - Network oriented
  - Strong web services support
    - Globus Toolkit 3
    - JAX Pack, Apache, etc.

How Does It Differ?

- Object heirarchy to hide complexity
- Simplifies error handling
  - Exception hierarchy
- Uses name space scoping
  - No more DRMAA_*

Implementation

- Adaptation of 1.0 DRMAA spec
- Built with API/SPI
  - Uses factory class to abstract vendor implementation
  - Up to vendor how to access DRM
- Default vendor implementation
  - Natively calls DRMAA C binding
- Borrows style elements from other APIs

What It Is Not

- Finished (Started)
- A way to submit Java objects
  - Jobs are still command line
    - `java com.sun.MyClass param1 param2`
  - VM startup overhead
- OGSI

Timeline

- 6/03 – Draft API Created
- 9/03 – Binding Document Created
- 10/03 – GGF9

- Depends on C binding
- C binding depends on spec
- Spec should be final

Web Services Binding

- Theoretically trivial
  - Java binding -> JAX-RPC -> WSDL Binding
- In reality more difficult
  - Proceedural -> OO -> Proceedural
  - Better to start with C binding
    - Tools aren't as clean
- Possibly easier to code by hand

Daniel Templeton

dan.templeton@sun.com