Steering and Service-based Visualization

Workshop “Computational Steering on the Grid” – Monday, May 7th 2007

NEC Corporation – HPC Marketing Promotion – Technology Solution

Pascal Kleijer

Why's & What's

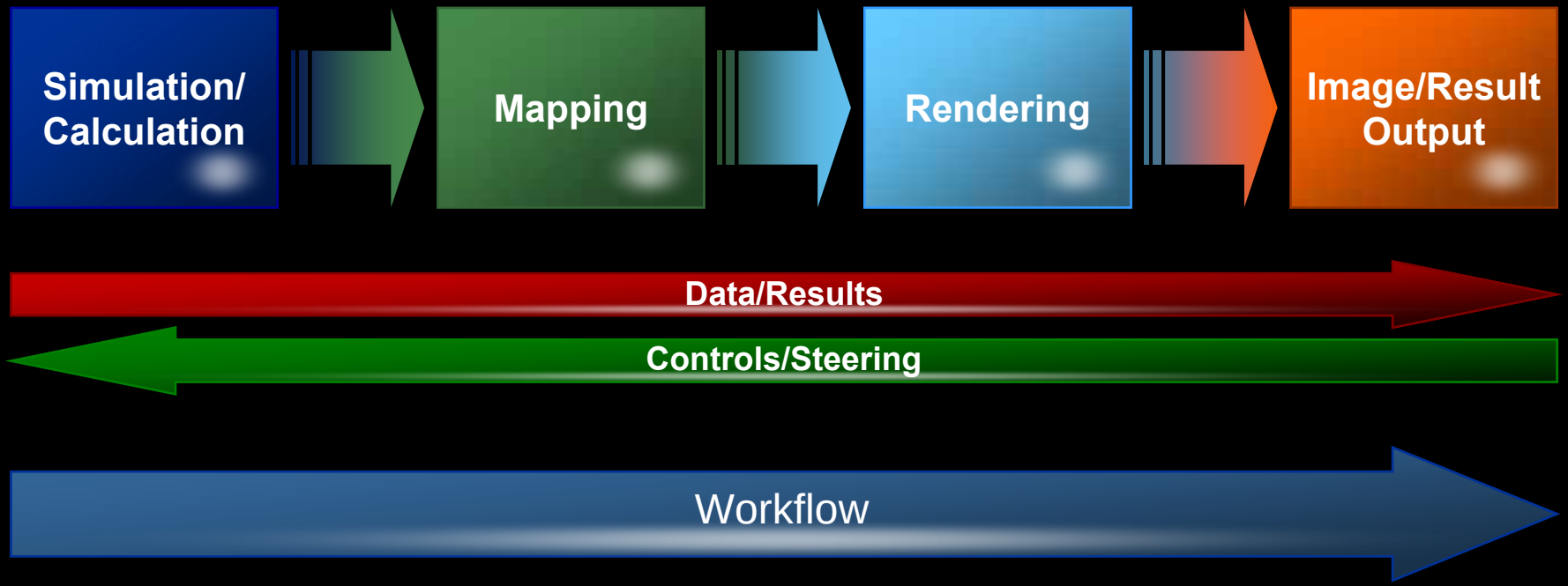
- Why do users want ***HPC***?
  - To perform simulations as fast and accurately as possible…
    → The results can be very *large-scale and complex*
- How do they obtain them?
  - By changing/tweaking the simulation parameters

➢ ***Steering*** is an important task for ***HPC*** users!

- What do they do with such results?
  - *Analyze* them…
- How do they deal with *human-unfriendly* numerical results?
  - Graphically ***Visualize*** them…
    Then, *recognize* and *interpret* them ***by Seeing***

➢ ***Visualization*** is an important task for ***HPC*** users!

NEC


Simulation/
Calculation
Mapping
Rendering
Image/Result
Output
Data/Results
Controls/Steering
Workflow

NEC

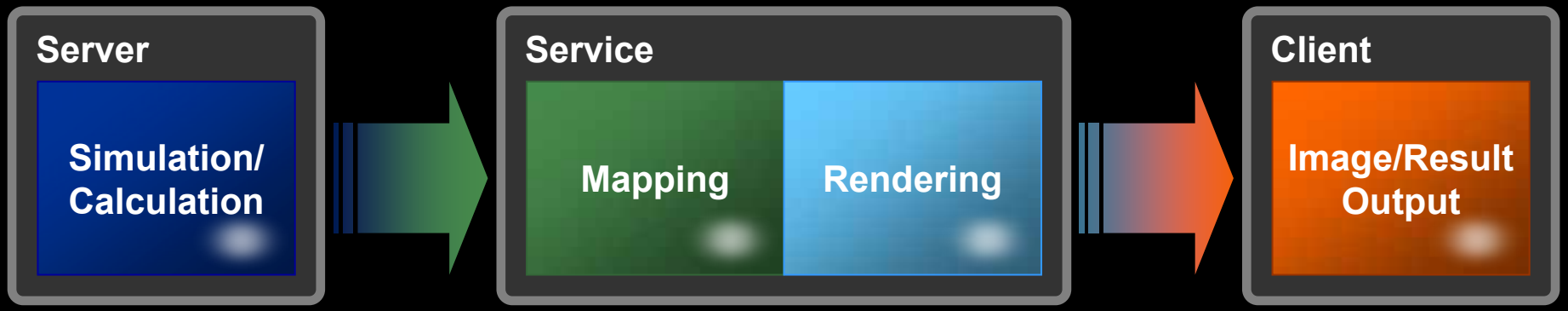
Approaches

Remote Rendering

- Pros
  - No need to down scale data
  - Solely uses existing hardware
  - Only the image size and encoding influence network load
  - Concurrent and Post-Processing Visualization
- Cons
  - Lower interactivity
  - Uses computational power for rendering

NEC

Approaches

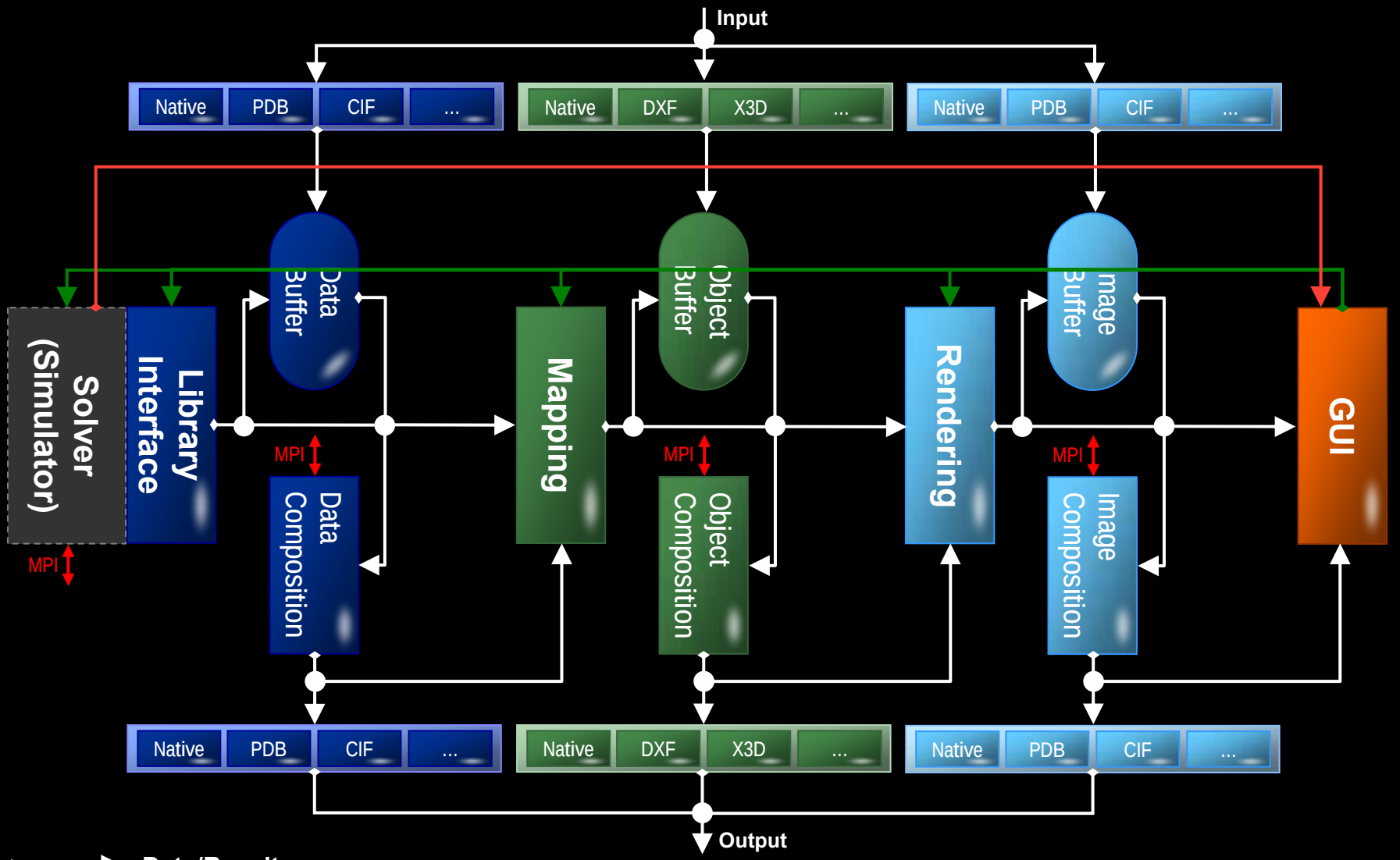


Service Rendering

- Pros
  - Distributed services
  - Can uses dedicated hardware
  - Can easily support multi-users
- Cons
  - Lower interactivity
  - More complex infrastructure
  - Network bottleneck (if data movement)

NEC

Flow Machine



NEC

RVSLIB

Real-Time Visual Simulation Library

Computational Resource

Internet / Intranet

User Terminal

Server

RVSLIB/Server

Client

RVSLIB/Client

Service

RVSLIB/Service

HDD

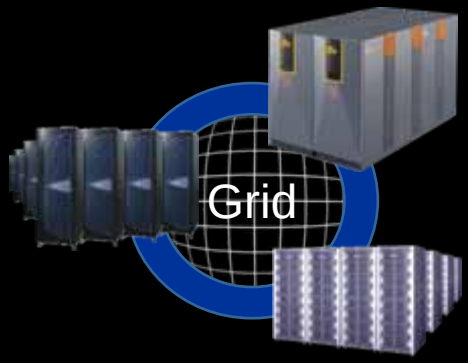
Feasibility Start R1.0 R1.1 R2.1 R3.1 R4.1 R5.1

'91 '92 ... '97 '98 '99 '00 ... '03 ... '07 ...

NEC

GVS

Grid Visualization System



Computational Resource | **Internet / Intranet** | **User Terminal**

Server

GVS/Visualizer

Client

GVS/Client

Service

GVS/Provider

Start → β1 → β2 → v1.0

'03 → '04 → '05 → '06 → '07 → …

NAREGI NII

NEC

Steering Past

- Commercial Product with Full Steering since ’97
- Simple Proprietary Protocol
  - Platform independent encoding
- Mapping/Rendering Steering is…
  - Binary with P2P
  - ASCII with Services
- Simulation Steering/Tracking is ASCII
  - Simple “Key-Value” pair model

NEC

Steering Present

- Shift to XML…
  - for storage
  - for service based communication
  - to increase description complexity
- Rendering…
  - inspired by OpenGL semantic but with proprietary XML
- Mapping…
  - proprietary XML
- Simulation…
  - proprietary XML based on “Key-Value” pair model
  - Meta Definition + Actual Values

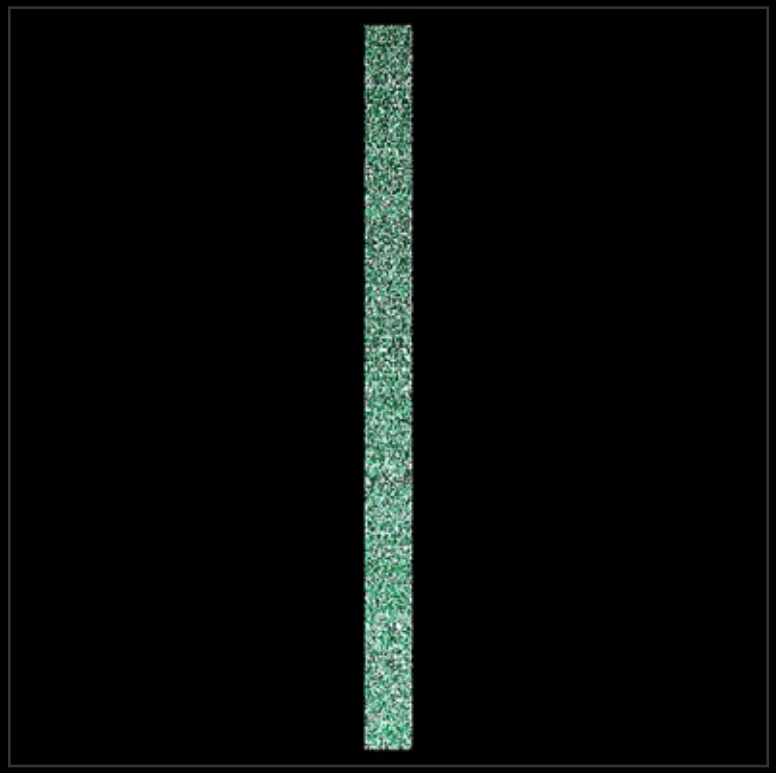
NEC

Steering Future

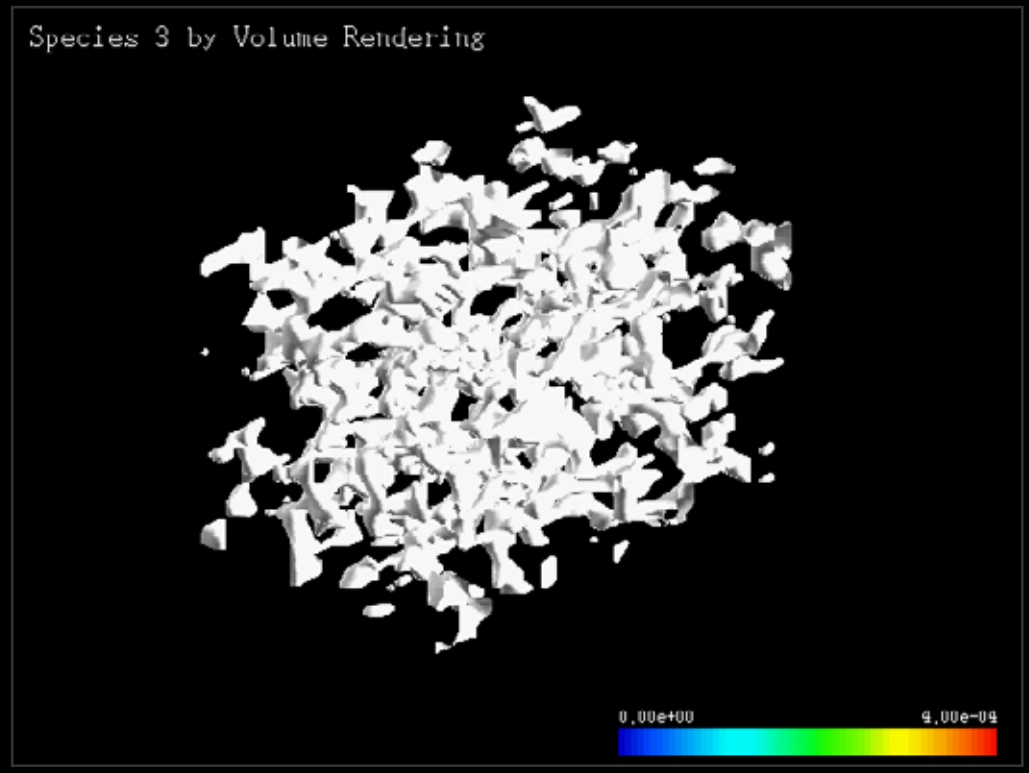
- XML for data
  - Use standards or create standards
    - Rendering: X3D, others…
    - Mapping: X3D, others…
    - Simulation: “Key-Value” pair model
- Transport should be format agnostic
  - OGF-SAGA Message API level
  - Alternate RPC-style approach
- Better Interoperability
  - GVid, eViz, RealityGrid, etc…

NEC

“Steering is Controlling,,
“Seeing is Believing,,



Oil Membrane, 20 million atoms
Data courtesy: Toyota Central R&D Labs, Inc.



Porous Media with Lattice Boltzmann
Data courtesy: C&C Research Labs, Europe

NEC

Empowered by Innovation