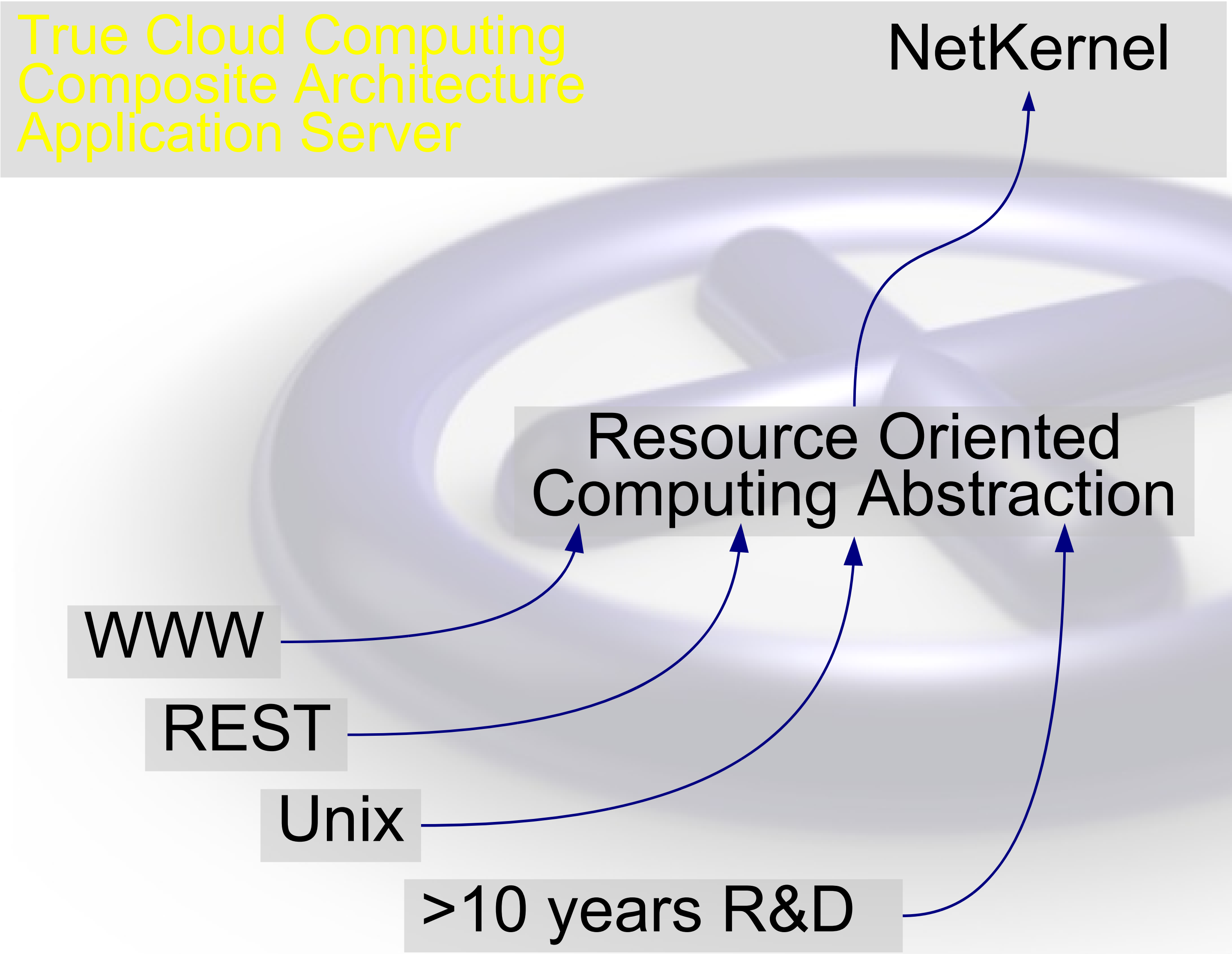
# NetKernel® and the Resource Oriented Cloud

# What is NetKernel?



True Cloud Computing
Composite Architecture
Application Server

NetKernel

Resource Oriented
Computing Abstraction

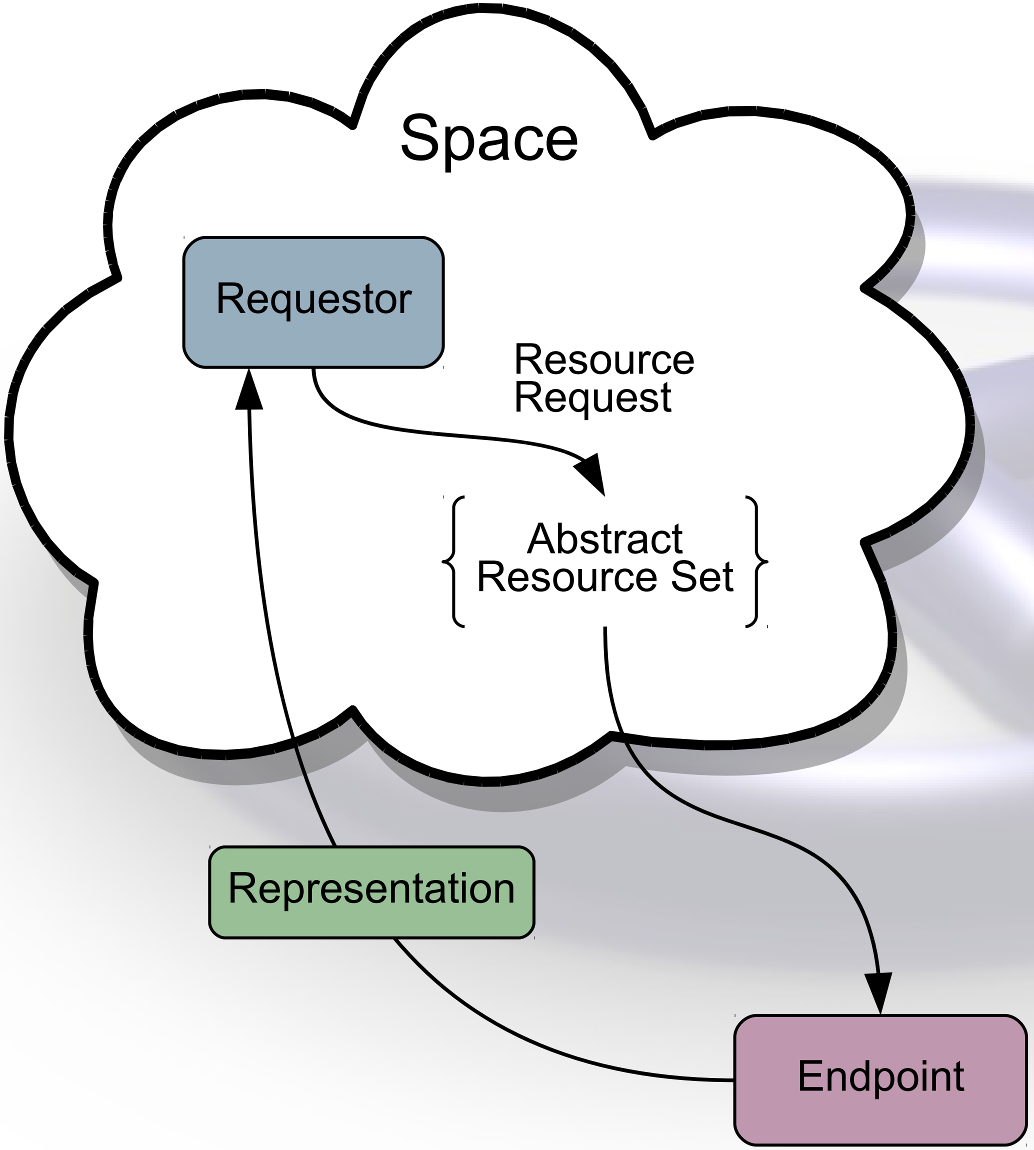
WWW

REST

Unix

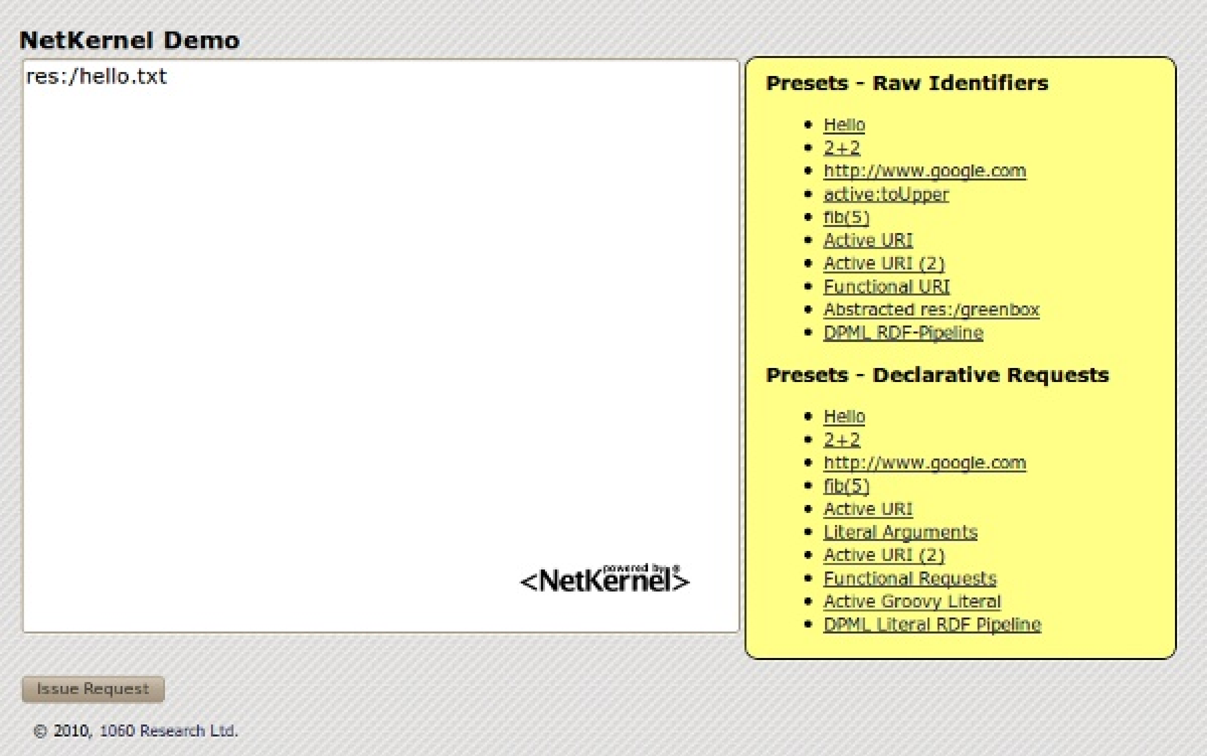
>10 years R&D

# Resource Oriented Abstraction



# Resource Requests

## Demonstration of ROC Concepts



**NetKernel Demo**

```
res:/hello.txt
```

**Presets - Raw Identifiers**

- Hello
- 2+2
- http://www.google.com
- active:toUpper
- fib(5)
- Active URI
- Active URI (2)
- Functional URI
- Abstracted res:/greenbox
- DPML RDF-Pipeline

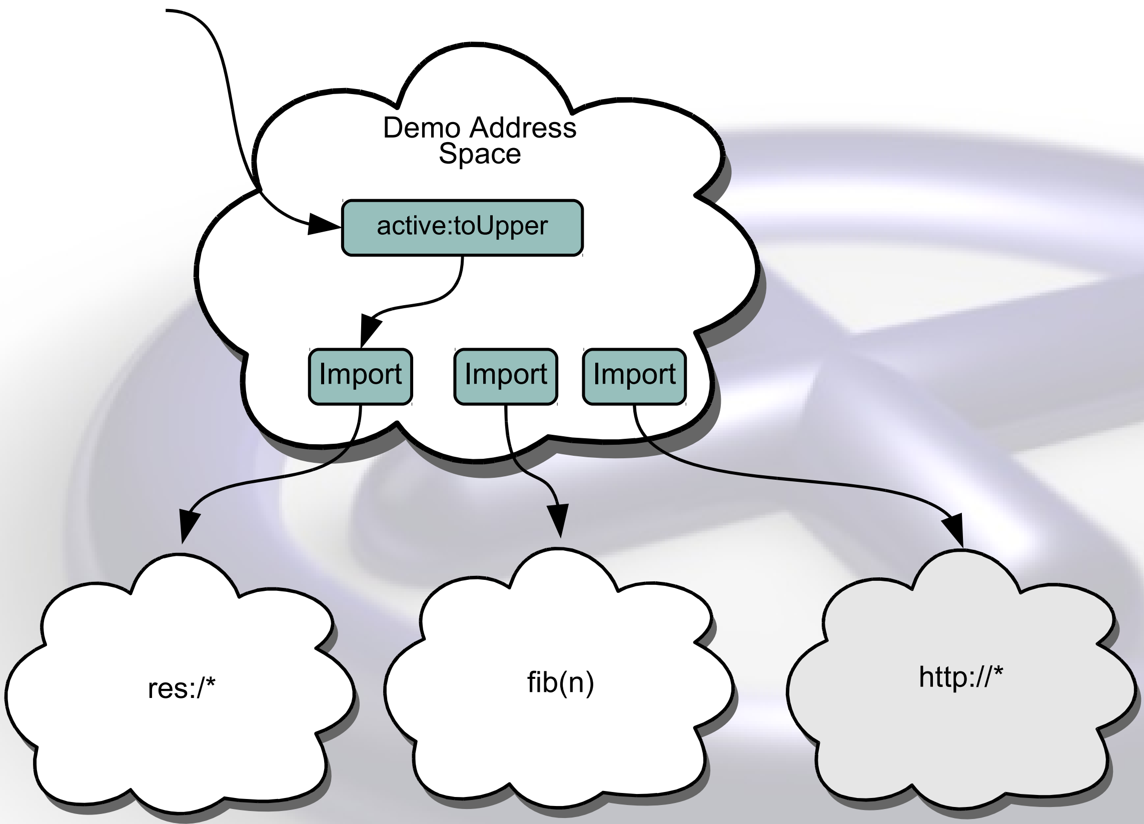
**Presets - Declarative Requests**

- Hello
- 2+2
- http://www.google.com
- fib(5)
- Active URI
- Literal Arguments
- Active URI (2)
- Functional Requests
- Active Groovy Literal
- DPML Literal RDF Pipeline

© 2010, 1060 Research Ltd.

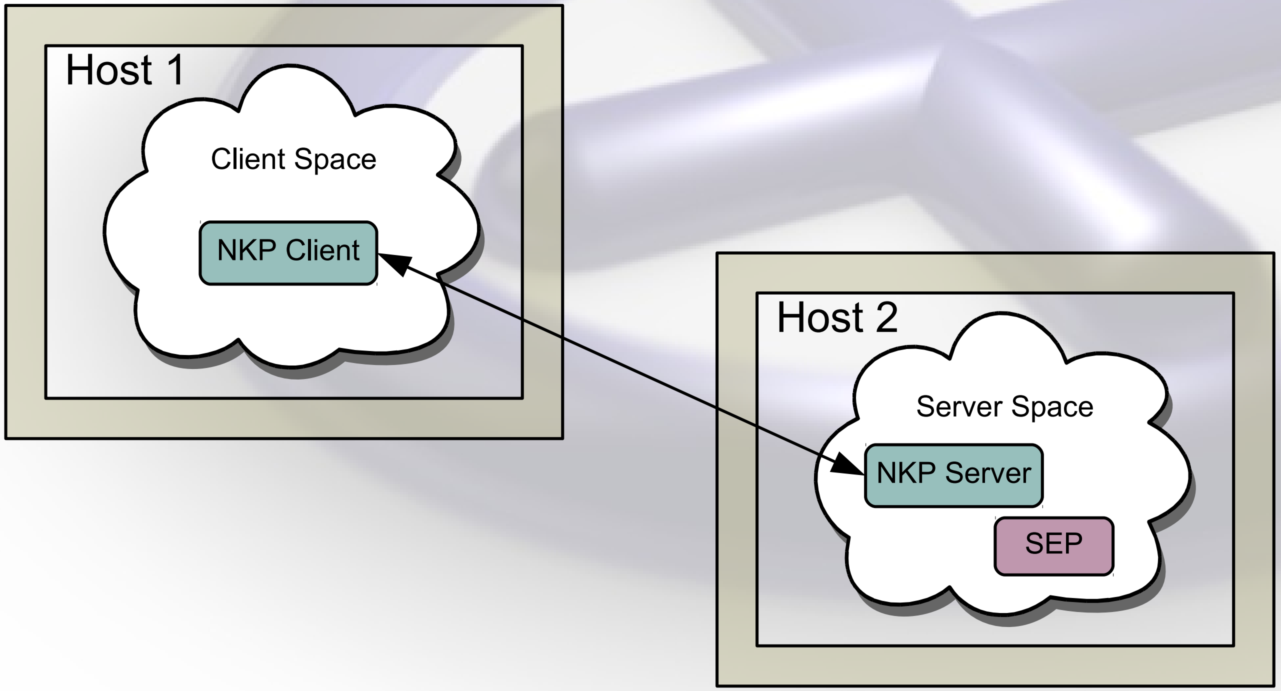
*In NetKernel Apposite install "demo1"*

# The Web *Inside*



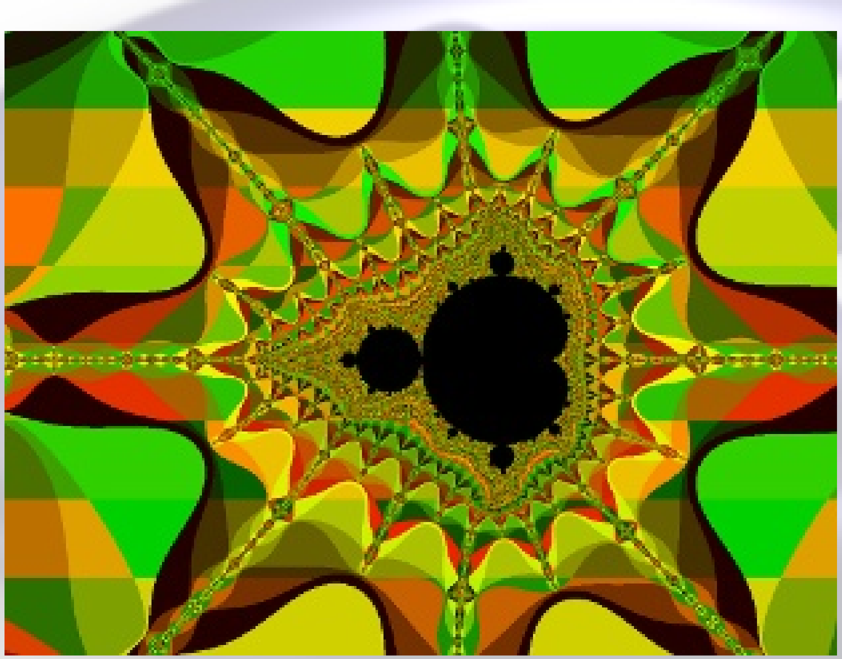
# ROC *Outside*

- NetKernel Protocol (NKP)
- Enables NetKernel ROC abstraction to seamlessly span hosts



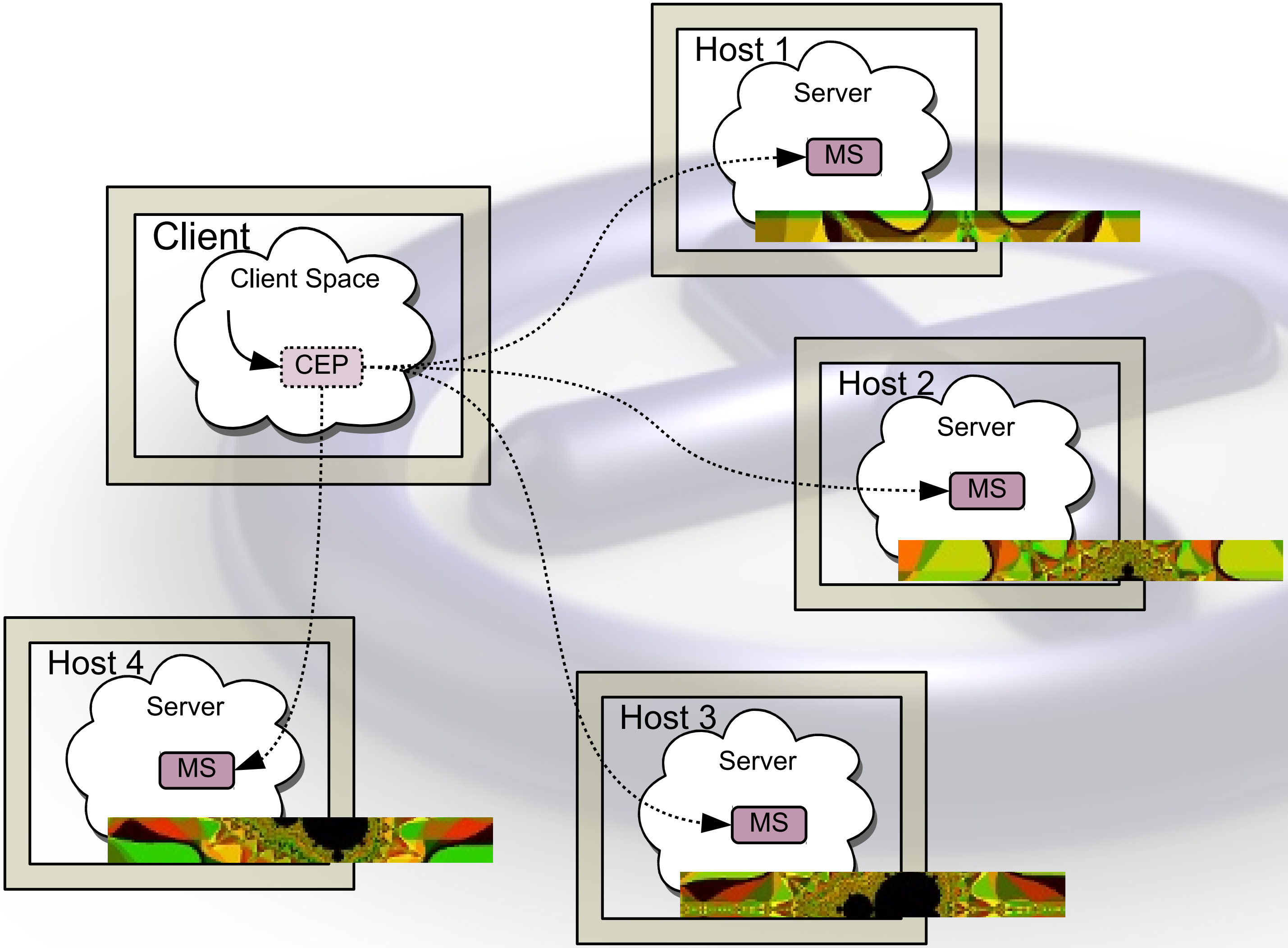
# NKP Demo

- Resource Oriented Mandelbrot Set
- active:mandelbrotStripe

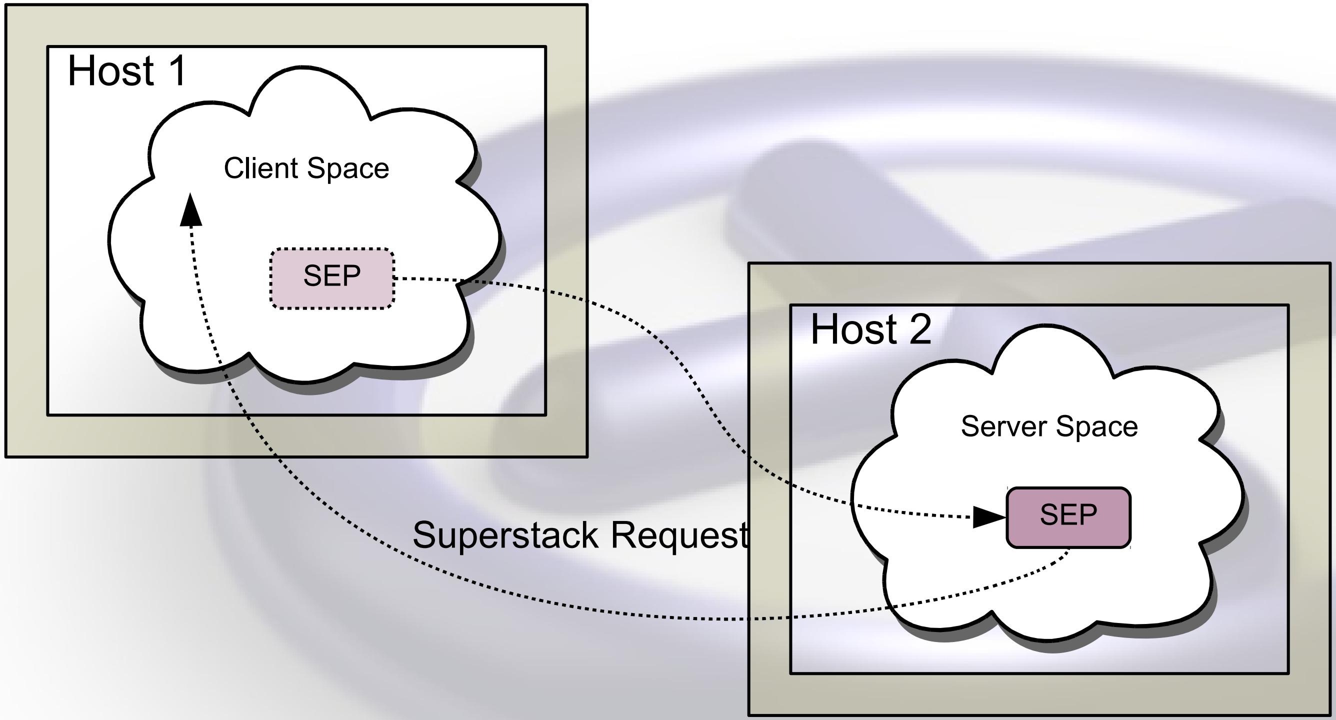


- Async fan-out of stripe computation
  - Scales Locally
  - Scales Remotely

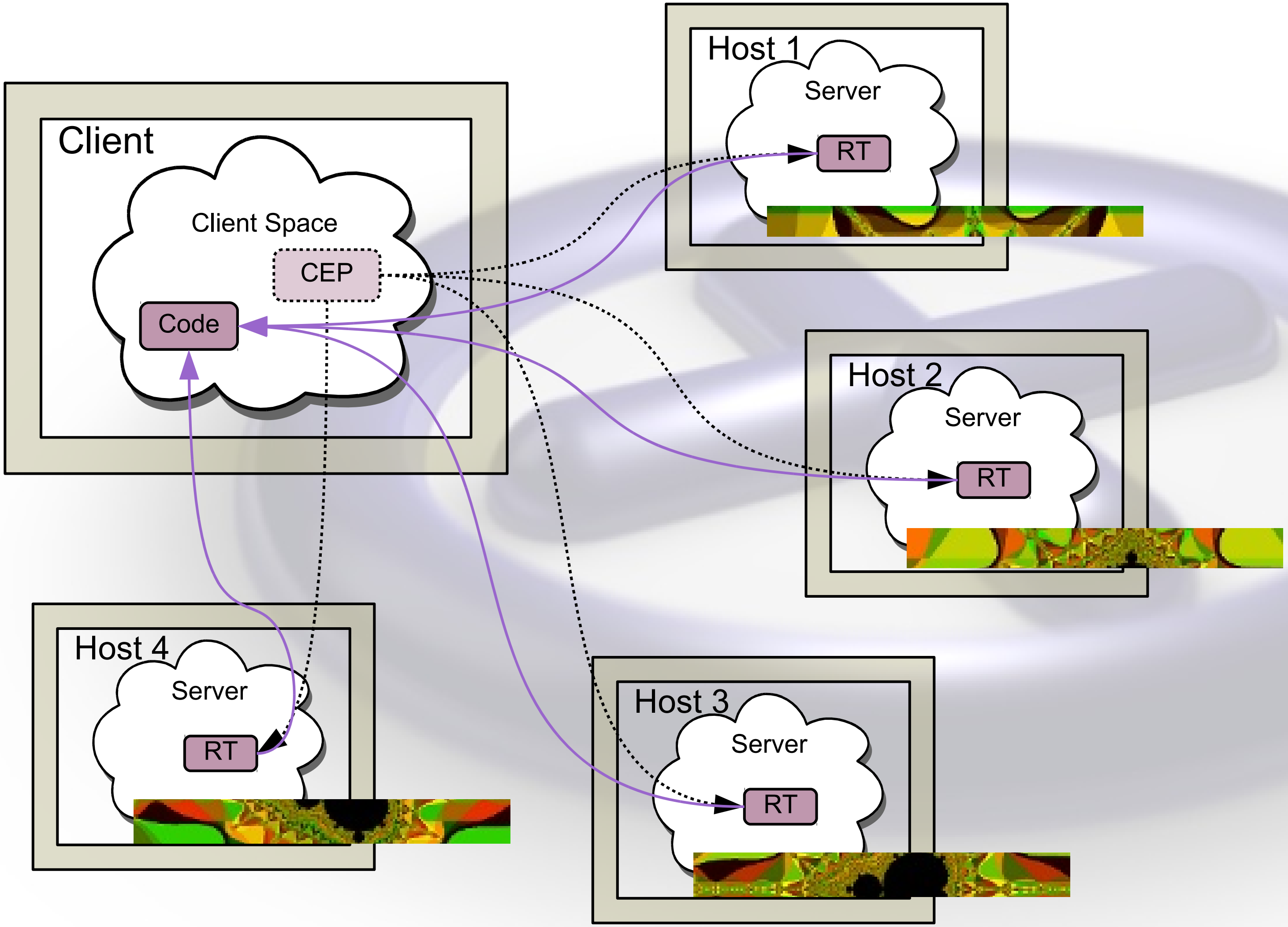
# NKP – Balanced OneShot



# Client-Server Inversion



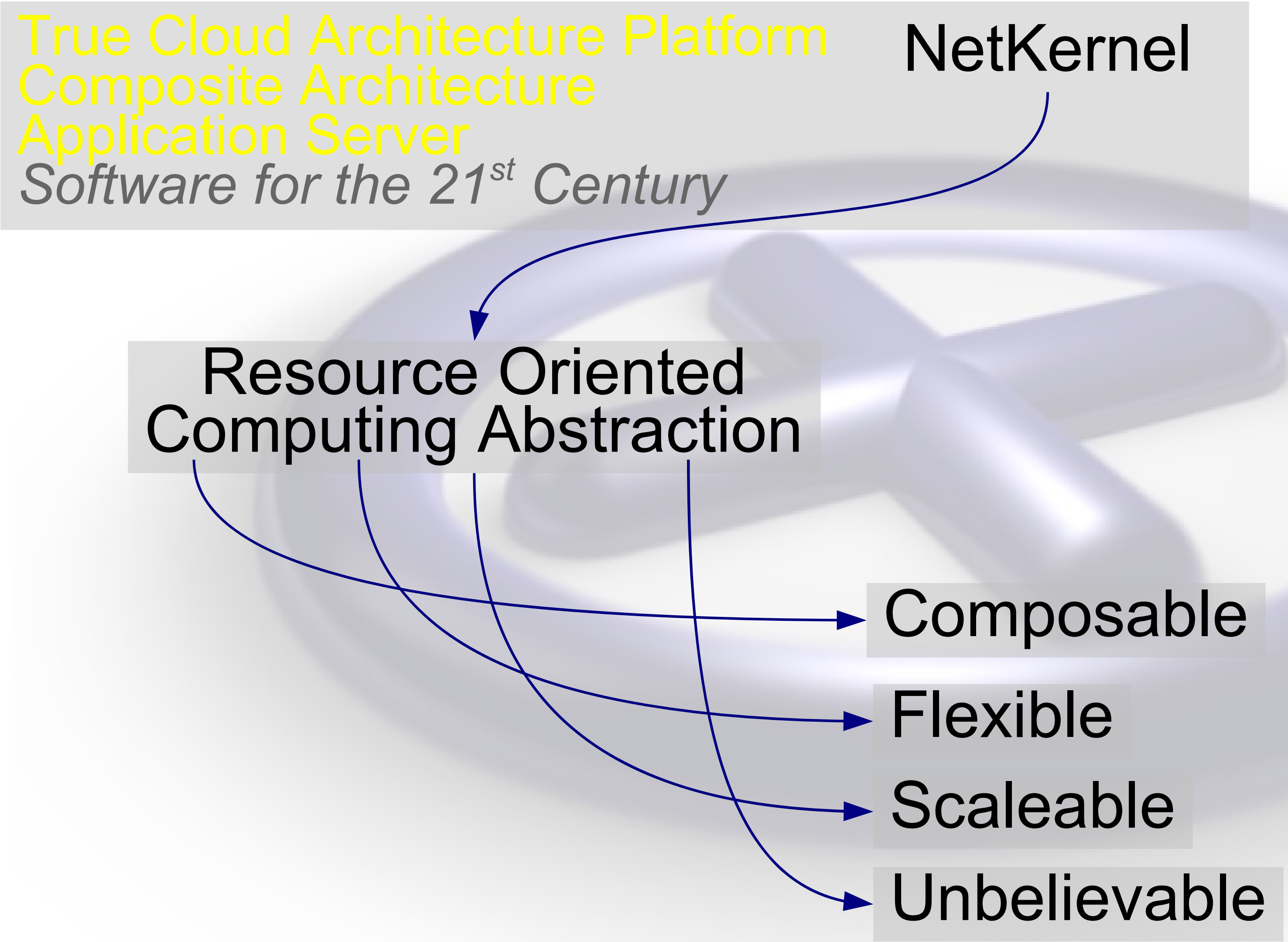
# NKP – Runtime Cluster



# NetKernel Protocol Features

- Asynchronous / Threadless design on both Client and Server
- Throughput Management via Async Throttles
- Pluggable Load Balancing Algorithms
- Oneshot Mode: REST++
- Mount Mode: NFS++
- Dynamically Reconfigurable Client
- Symmetric or Asymmetric Spacial Scope (Superstack)
- Architectural Trust Boundaries
- Distributed Caching and Dependency Model

# Why NetKernel?



True Cloud Architecture Platform
Composite Architecture
Application Server
*Software for the 21st Century*

NetKernel

Resource Oriented Computing Abstraction

Composable

Flexible

Scaleable

Unbelievable