

REST AND THE REST

Ajax

- * First step into asynchronous world
- * Content could change without page reloading
- * The web eventually becomes interactive
- * The client drives the webapp state



Web 2.0

- * Users are the biggest content creators
- * Exponential data growth
- * Distribution is the only option
- * Statelessness prevents complexity



IoT - WoT

- * Everything can produce and consume data
- * Unknown devices communicate with unknown servers
- * Now, this is anarchic scalability

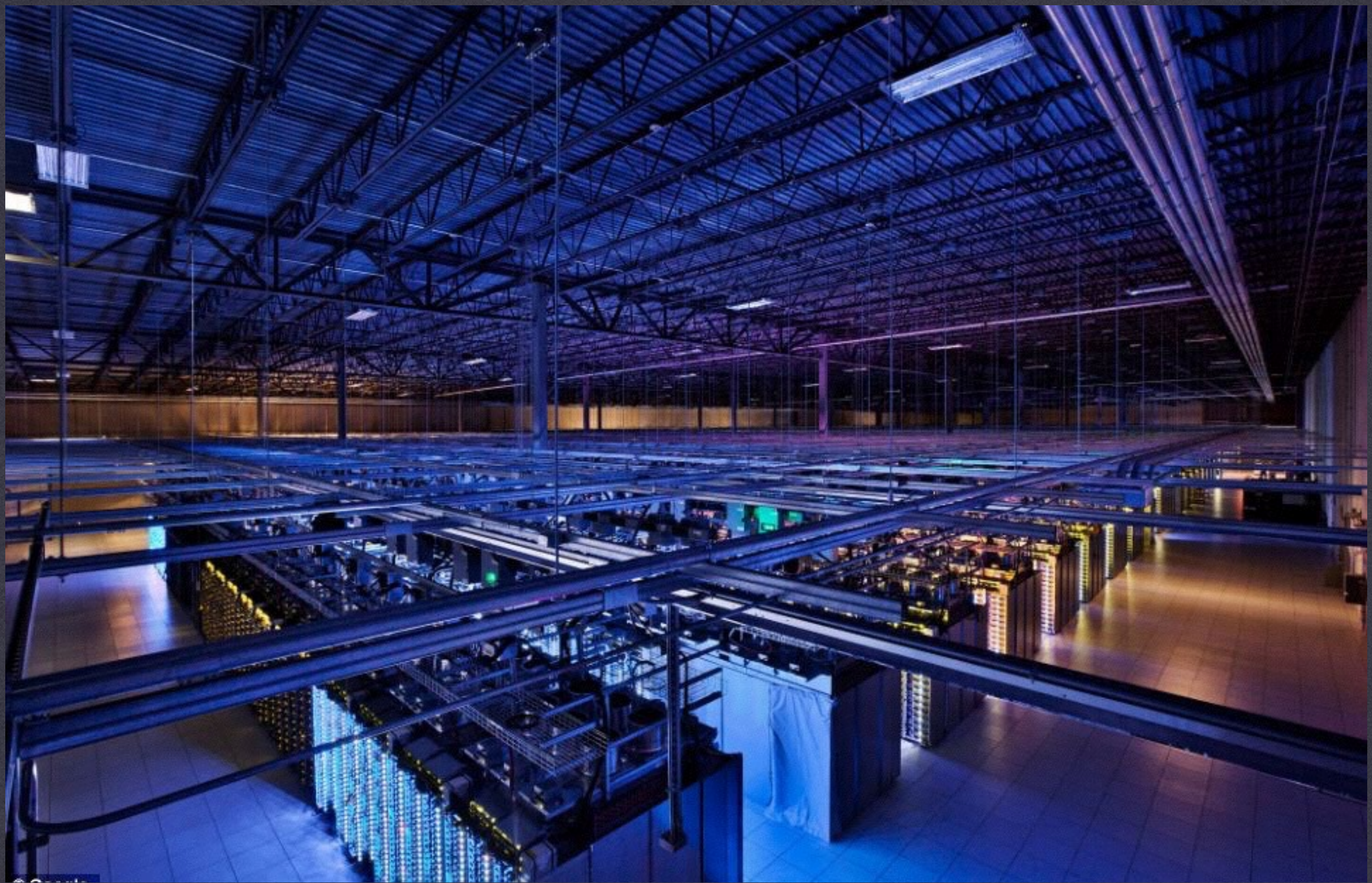


Samsung
GALAXY

Cloud

- * Computation power becomes commodity
- * Scalability and performance are key
- * Resources can be dynamically allocated and claimed
- * The scenario Berners-Lee depicted but on steroid





HOW REST IS HOLDING UP?

“Those who cannot remember the past are
condemned to repeat it”

–George Santayana

Let's get back to requirements

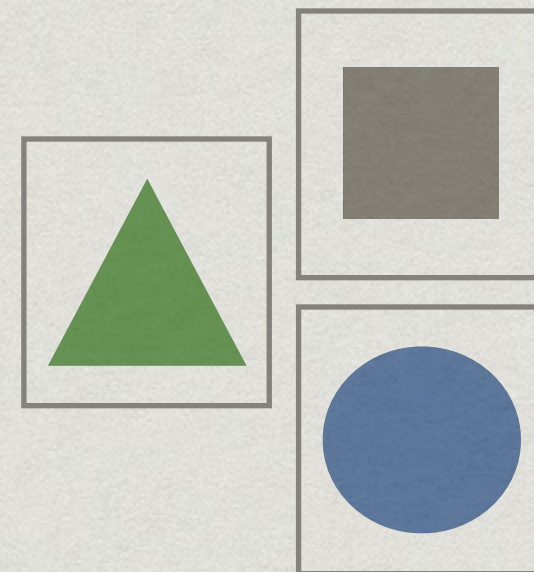
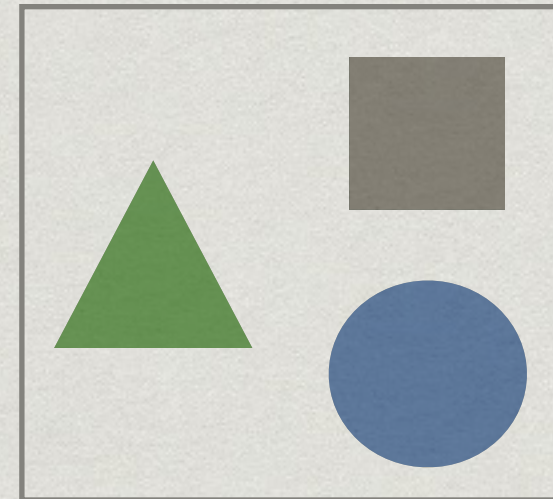
- * Low Entry Barrier: simple protocol and languages, easy to discover new features
- * Extensibility: add new bit of data/features
- * Heavy distribution: partial knowledge, slashdot effect, independent deploys

Microservices

REST

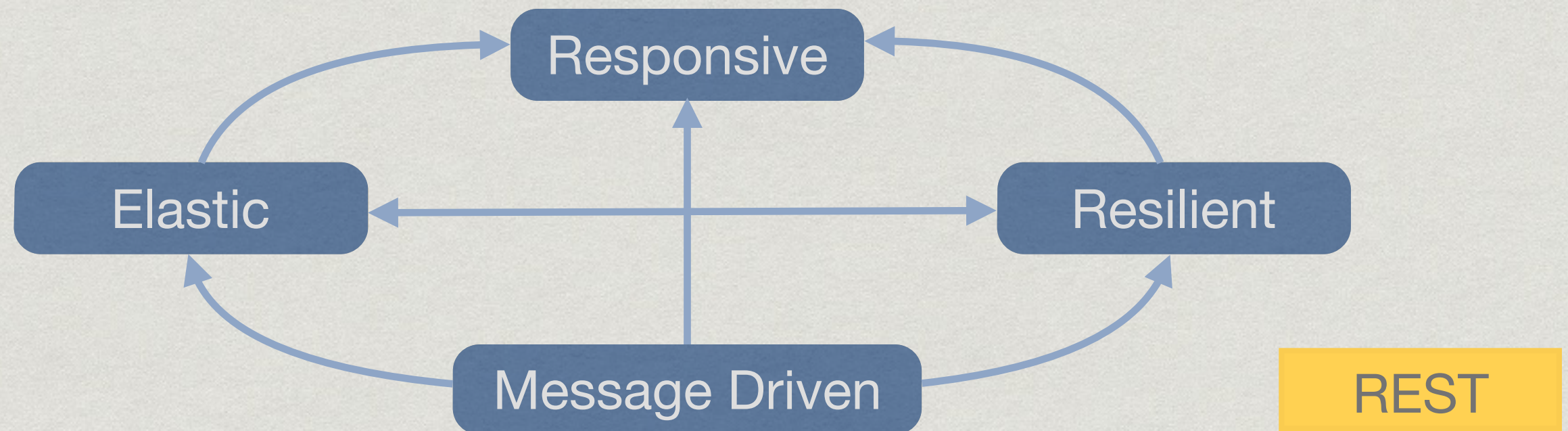
Monolith

- * Componentization via Services
- * Organized around Business Capabilities
- * Products not Projects
- * Smart endpoints and dumb pipes
- * Decentralized Governance
- * Decentralized Data Management
- * Design for failure
- * Evolutionary Design



Micro Services

Reactive applications



- * Responsive: responds quickly and detects problems early
- * Elastic: can scale up and out
- * Resilient: it doesn't fail, it handles errors nicely
- * Message Driven: asynch communication to foster decoupling and enable previous properties

Summary

- * The Good: REST is holding up to scale and distribution
- * The Bad: Its client-server focus is struggling with asynchronicity
- * The Ugly: people ignore its basic design principles —> Cargo Cult design

