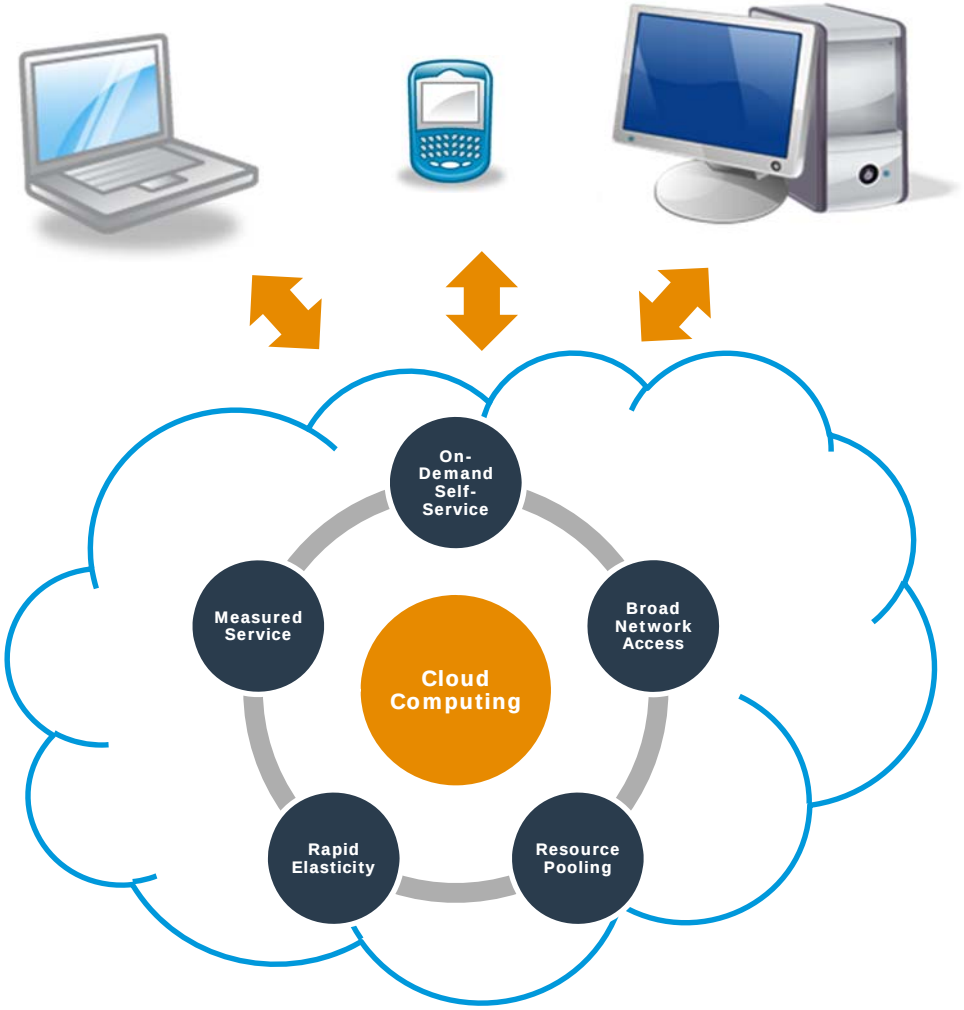
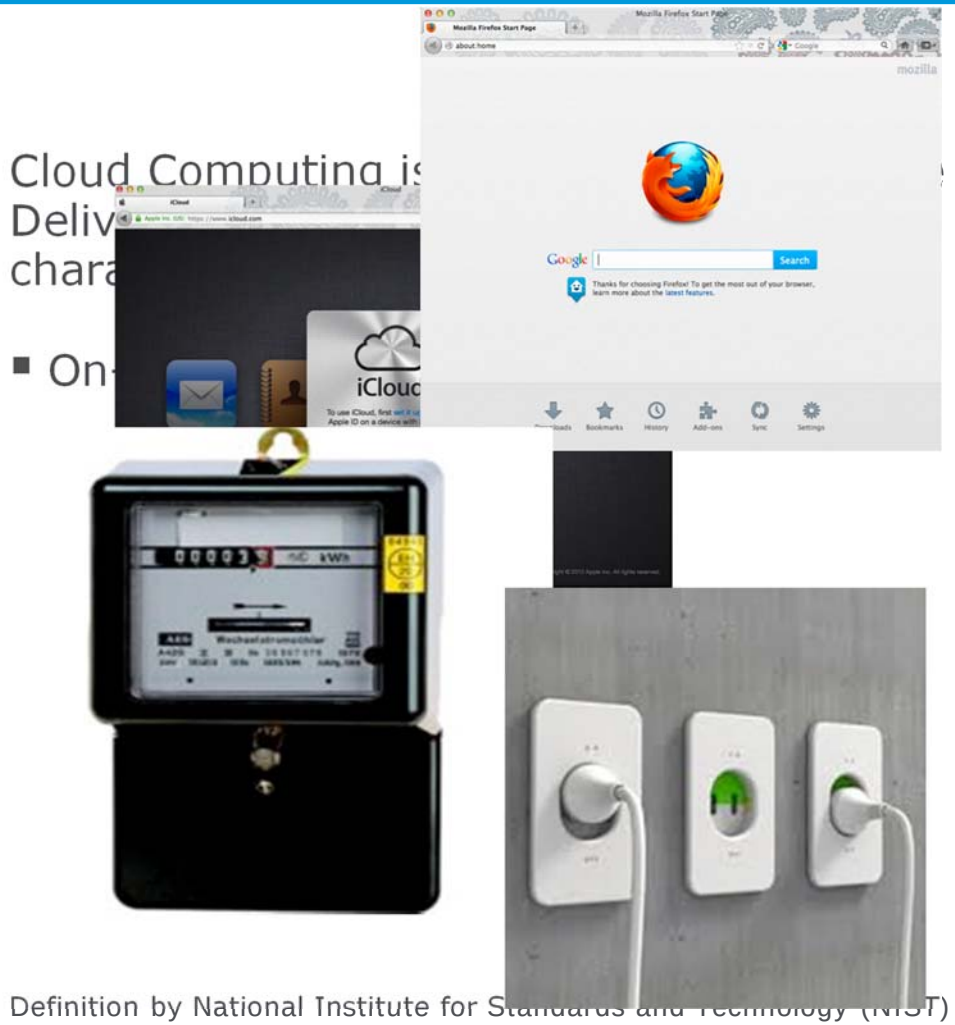
Cloud Based Architectures in Ground Systems of Space Missions

M. Sarkarati, M. Merri, M. Spada
HSO-GDA
ESA/ESOC

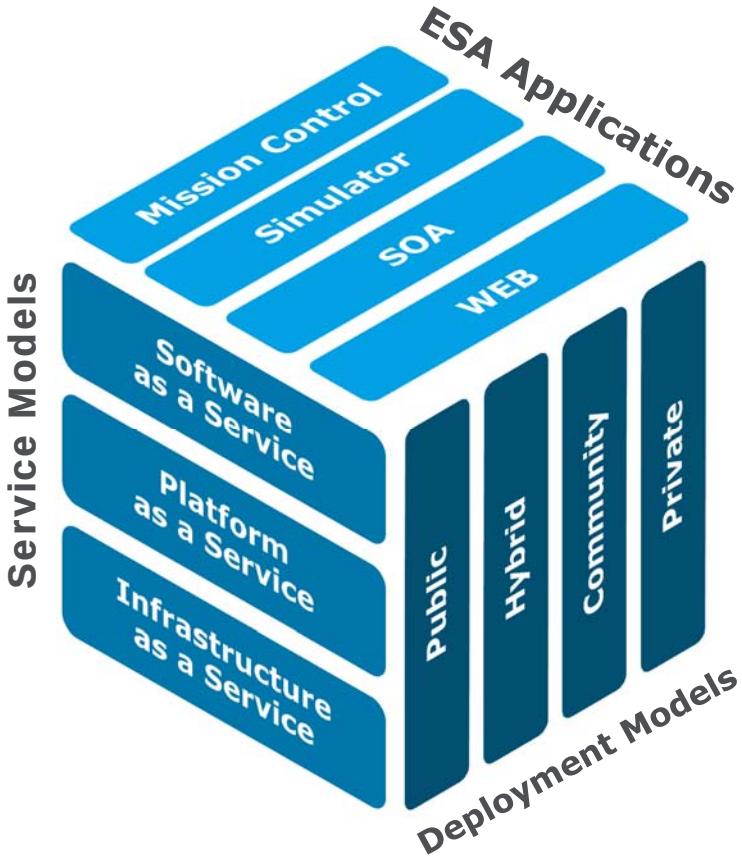
Copyright

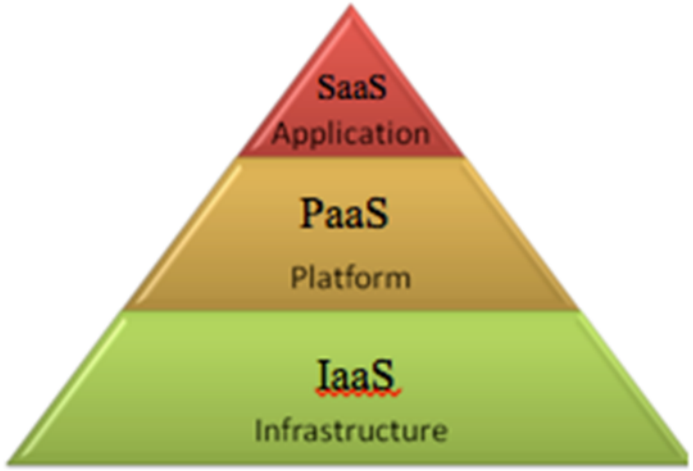
Common Understanding Of Cloud Computing





Which Cloud?





- Public
- Private
- Hybrid
- Community

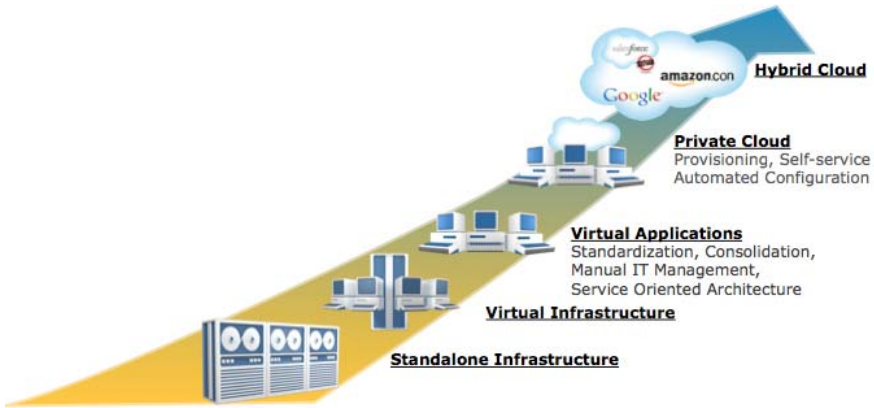
** Definition by National Institute for Standards and Technology (NIST)*

Challenges Of Modern Ground Data Systems

- **Distribution** and **location independency** of ground data systems
- **High Availability** of ground segment services to the user community
- **Scalability** and increasing demand on computing resources
- Utilisation of e2e **Off-The-Shelf solutions** for common IT tasks
- **Federation** of disperse solutions, System of Systems (**SoS**)
- Moving towards Service Oriented Architectures (**SOA**)
- **Security** and risk management awareness
- **Baseline Management** of a large number of systems

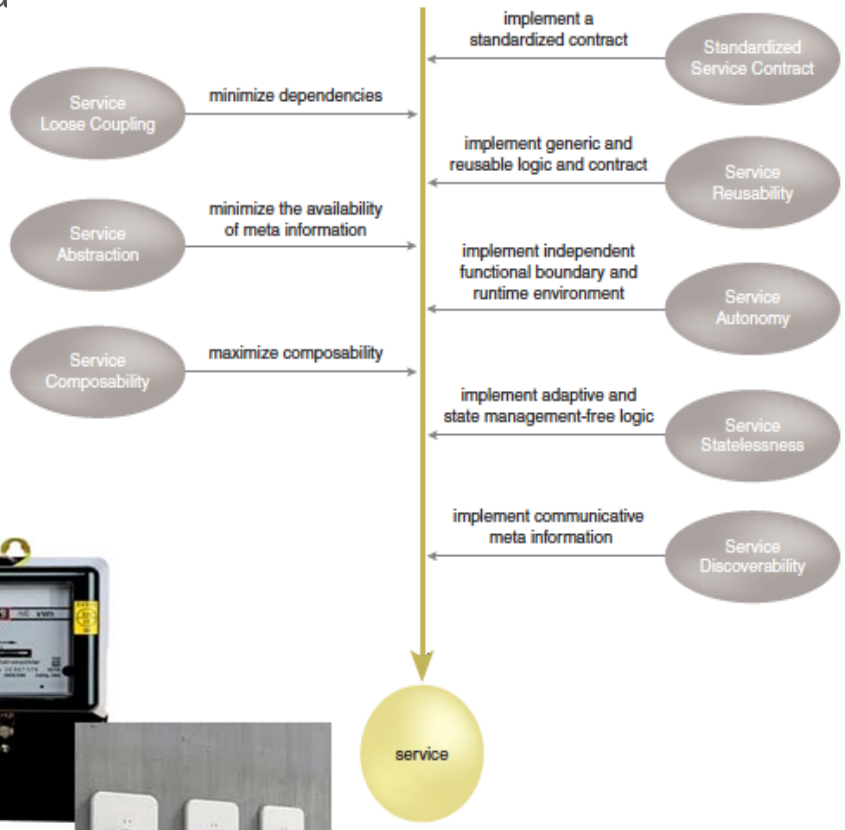
Cloud As An Architectural Design Paradigm

- Cloud computing is not just a technology. It is an architectural design paradigm
- CC can often not be applied on top and must be built into the architectural design
- The higher you go in the service model pyramid (PaaS/SaaS) the more this is true
- Basic principle of Simple Design
  - Some sources of complex design delegated to lower level cloud services
  - Performance and optimisation
  - Multi-threading, caching, session management
  - Failure tolerance, Redundancy Load Balancing
- Design to run on a Cloud Platform
  - Google App Engine, SalesForce.com
  - AWS: design and deployment (work flows, …)
  - Hadoop HDFS and Map-Reduce
  - Google Bigtable and Amazon SimpleDB



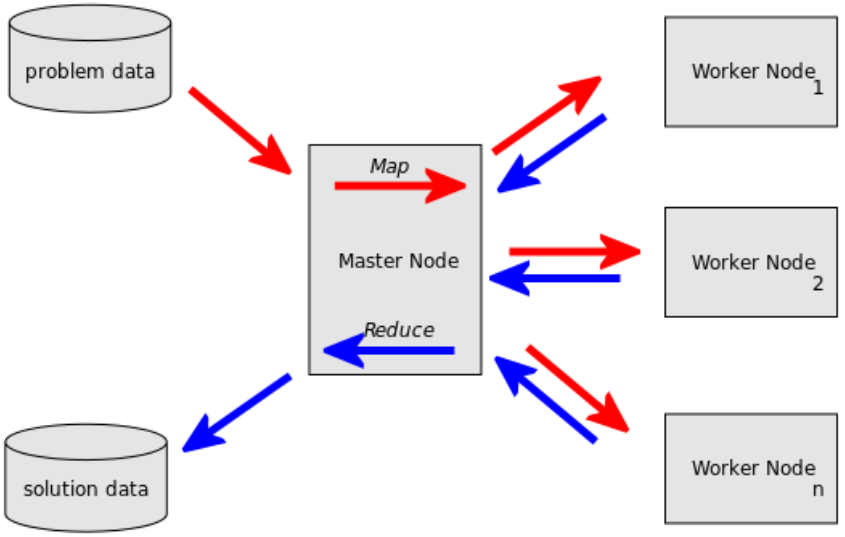
Design Principles For The Cloud

- Net-Centric Communication
  - Don’t use File based or inter-process communica
- Principles of service oriented design (SOA)
  - Loose coupling
  - Standardised service contracts
  - Statelessness
  - Autonomy
  - Abstraction
  - Discoverability
  - Reusability
  - Composibility
- Built-in security and virtual appliances
- Management and measuring interfaces
- Design to cost



Example Of The Impact On The Design: Hadoop In The Cloud

- Apache Hadoop Framework:
  - Distributed File System HDFS
  - Job Scheduling and Cluster Manageme
  - Automatic handling of node failures
- Map-Reduce:
  - Map(k1,v1) → list(k2,v2)
  - Reduce(k2, list (v2)) → list(v3)

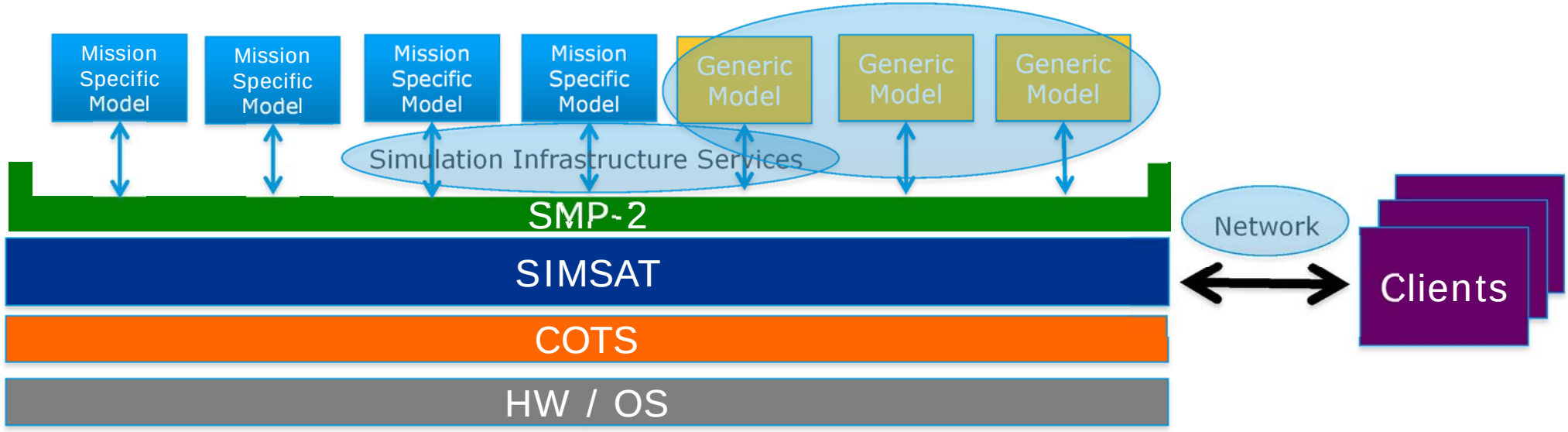


Source: http://en.wikipedia.org/wiki/MapReduce

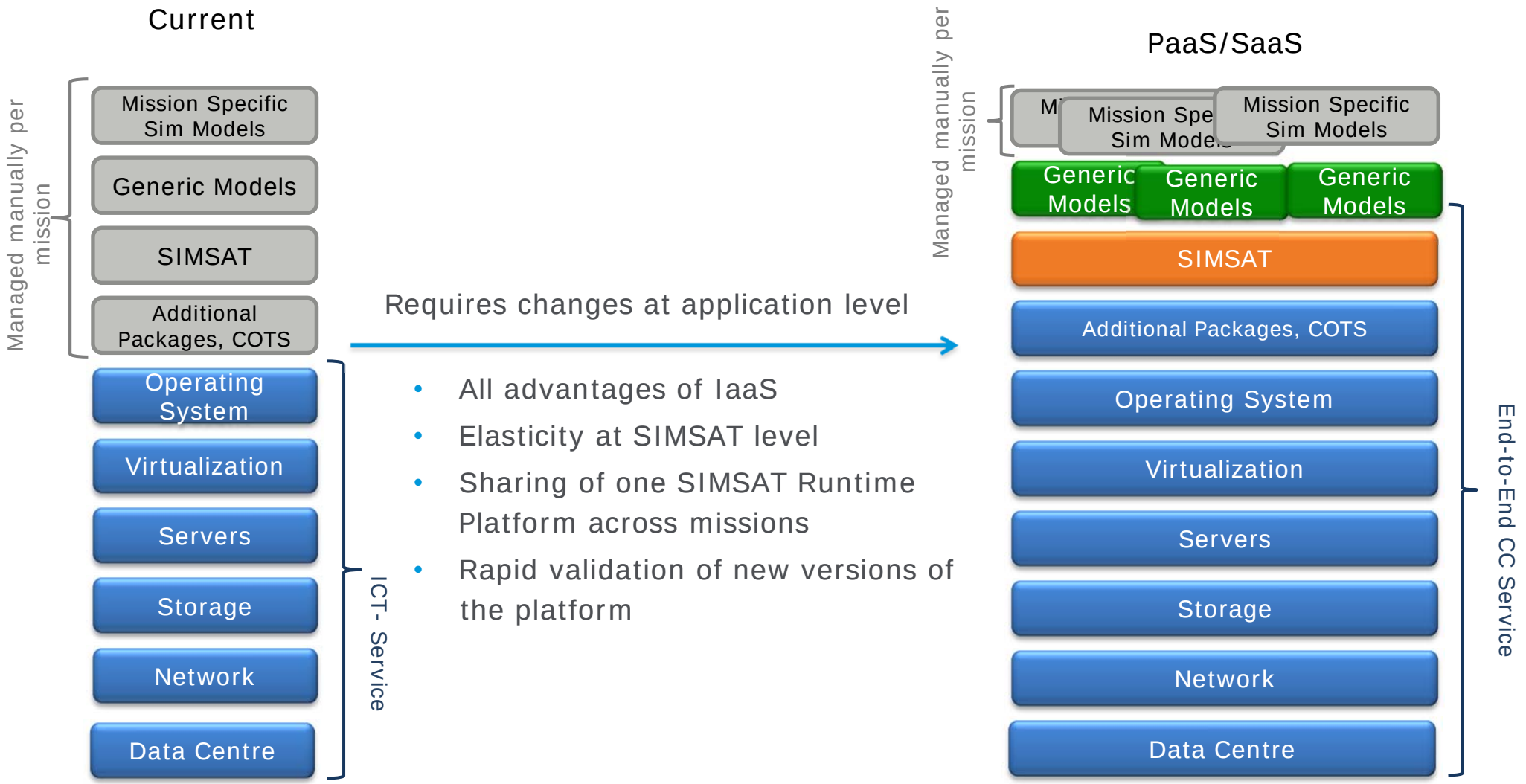
```
function map(String name, String document):
  // name: document name
  // document: document contents
  for each word w in document:
    emit (w, 1)

function reduce(String word, Iterator partialCounts):
  // word: a word
  // partialCounts: a list of aggregated partial counts
  sum = 0
  for each pc in partialCounts:
    sum += ParseInt(pc)
  emit (word, sum)
```

Transition To A Cloud Solution: The Case Of SIMULUS

- SIMULUS: Generic multi-purpose simulation platform
- Standardised Platform services (Scheduling, Logging, Eventing, Breakpointing …)
- Simulation Model Portability Standard SMP-2
- Generic models reusable cross mission in operational simulators
- Systematic cloudability analysis of select ground data systems in 2010-2011
- Candidates for each cloud service/deployment model identified



The Case Of SIMULUS
Step -2 : Evolution towards PaaS/SaaS



Transition To A Cloud Solution: The Case Of SIMULUS PaaS/SaaS

- For IaaS service model on a private cloud → No changes on SIMULUS side
- For PaaS and SaaS → Changes on SIMULUS design and implementation
  - New concepts for user management
  - New concepts for resource management
  - New concepts of workspaces and separated simulation sessions
  - New concepts for deployment of simulation models onto the platform
  - New concepts for data and file sharing
  - Enhancements to clients (Man-Machine Interfaces)
- Generic models → SaaS
  - New concepts for dependency configuration
  - New concepts for deployment of generic models

Take Aways

- Cloud Computing is not just a technology but an architectural paradigm
- Cloud Computing solutions expose certain common characteristics (NIST)
- Do not sell refrigerator to Eskimos (Apply Cloud where you can benefit)
- Cloud Computing can often not be added “on-the-top”
- It must be built in the Architectural design of the system
- Moving a legacy system to Cloud does often require changes at some level
- The higher you go in the Cloud service model the more changes are needed
- Perform systematic cloudability analysis and pick the applications which
  - Are most suitable for a certain cloud model
  - Have the highest potential in benefiting from Cloud characteristics

Thank you

ESA/ ESOC
M. Sarkarati, M. Merri, M.Spada
Email: firstname.lastname@esa.int

Design To Fail

- Capability Decomposition → Autonomous, loosely coupled Services
- Net-Centric asynchronous communications (with queuing)
- Failure isolation → ICT Failure in one node does not propagate to other nodes
- Fast Recovery and redundancy provided by the cloud
  - Fully automated provisioning of services
  - Services must be able to re-configure automatically and join the application
- Statelessness and state deferral
- Much lower impact of failure as only limited nr of users are impacted