Leveraging Domain-Specific Software Architectures for Classifying Cloud Service Abstractions

T.S. Mohan[1] Nenad Medvidovic[2] Chris A. Mattmann[2, 3]

[1]*Ecom Research Labs*
Infosys Technologies
Bangalore, India
subramanian_mohan@infosys.com

[2] *Computer Science Department*
University of Southern California
Los Angeles, CA 90089, USA
{neno,mattmann}@usc.edu

[3] *Jet Propulsion Laboratory*
California Institute of Technology
Pasadena, CA 91109, USA
mattmann@jpl.nasa.gov

Have We Really Understood the Cloud Yet?

- Plethora of Definitions – Hype?
  - Range of Technologies and business models
- What really clicks in the Cloud?
  - Pay per use – no capex only opex!
  - Meet seasonal loads – elasticity – scalability up and down
  - Simplified uniform abstractions
- Service offerings and Vendors
  - Large players – Azure, AppEngine, AWS, etc
  - Small niche players – variety of cloud enablers for the varied enterprises.
  - Value add? Where and How? Sustainable?
- Key Issues and Challenges for sustained usage
  - Applying / Leveraging CAP (Consistency, Availability & Performance)
  - Security (or is it?) – The dilemma of storing and using Confidential Data
  - Interoperability and portability
  - Variable seasonal Cloud Services Pricing
  - Multi-tenancy and reputation sharing

The Users Dilemma – Migrating into the Cloud…..

IaaS
IT Folks

- Abstract Compute / Storage / Bandwidth Resources
- Amazon Web Services – EC2, S3, SDB, CDN, CloudWatch

PaaS
Programmers

- Abstracted Programming Platform with encapsulated infrastructure
- Google Apps Engine (Java/Python); Microsoft Azure (.NET)

SaaS
Architects & Users

- Application with encapsulated infrastructure & platform
- Salesforce.com; Hotmail;Gmail; Yahoo Mail; Facebook; Twitter

The Compromise…… Hybrid Clouds!

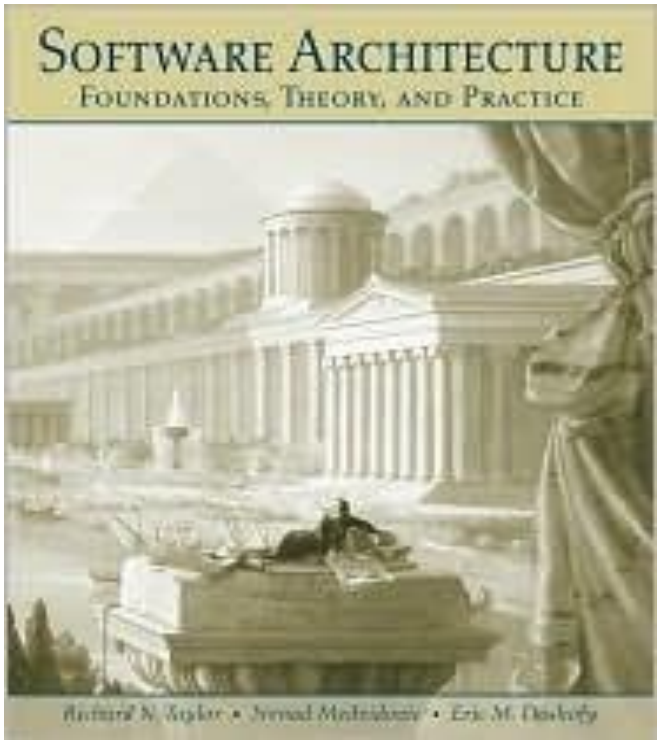
Leverage what……?

- At what level of the cloud services .. Combinations of IaaS, PaaS and SaaS.
- What costs model ? Factoring in variable pricing risks….
- What configurations and deployments? Interoperability? Portability?

Can we apply the principles and discipline of Software Engineering when using Cloud services?

Domain Specific Software Architectures

- A domain-specific software architecture (DSSA) comprises:
  - a reference architecture, which describes a general computational framework for a significant domain of applications;
  - a component library, which contains reusable chunks of domain expertise; and
  - an application configuration method for selecting and configuring components within the architecture to meet particular application requirements.
- *A Reference architecture* is the set of principal design decisions that are simultaneously applicable to multiple related systems, typically within an application domain, with explicitly defined points of variation.



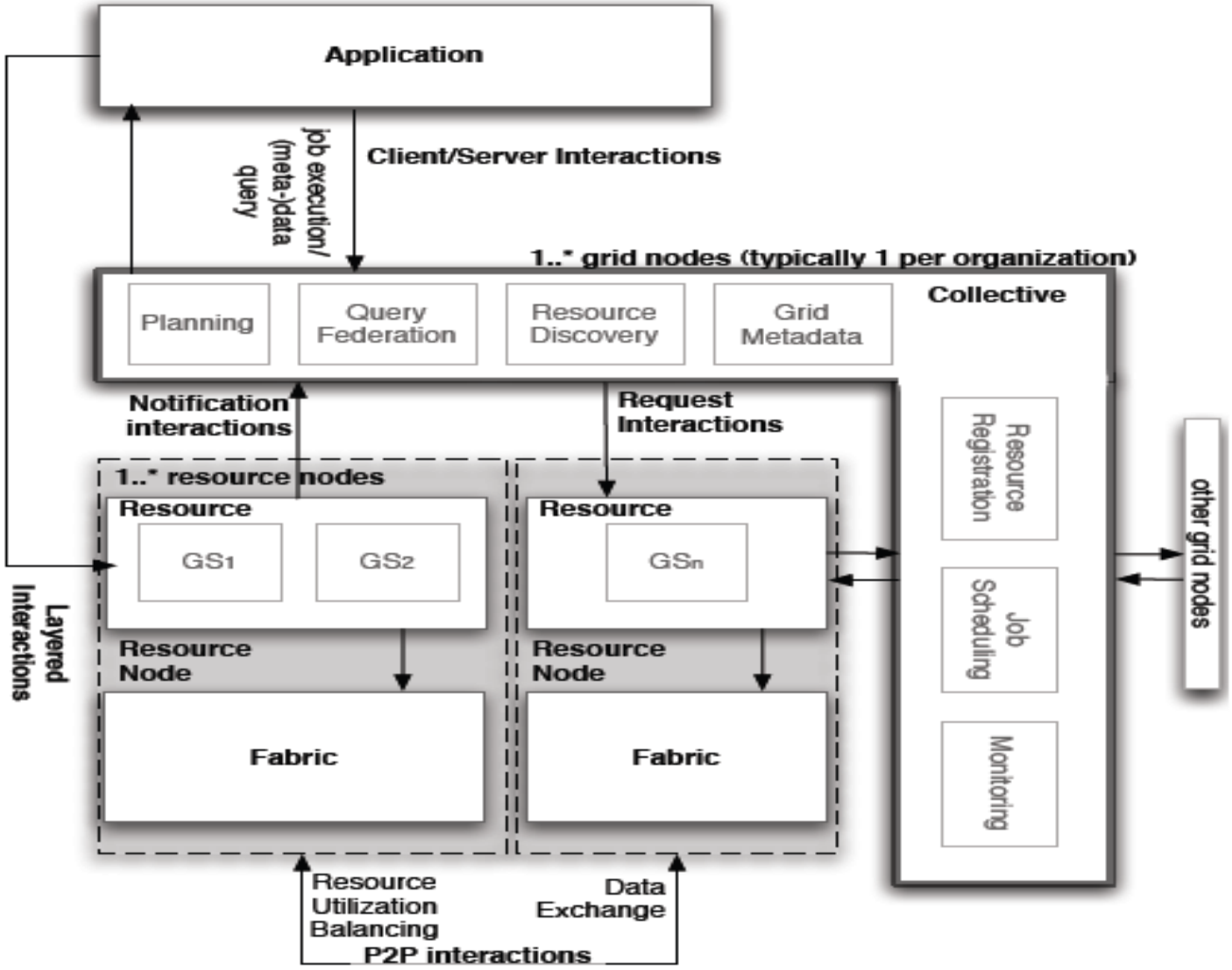
Leveraging Studies from the World of Grid Technologies

Table 1. The studied grid technologies.

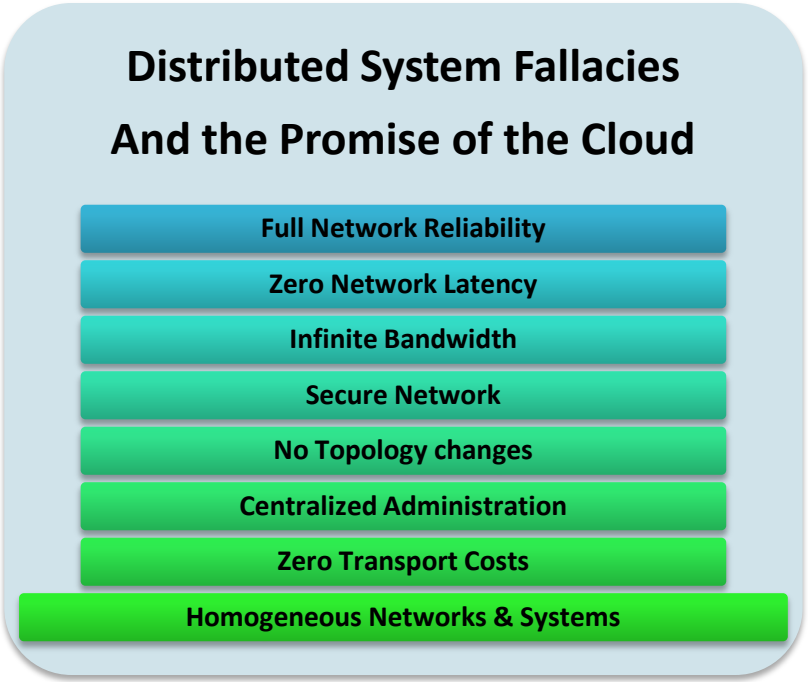
Technology	PL	KSLOC	#ofModules	URL
Alchemi	C# (.NET)	26.2	186	http://www.alchemi.net
Apache Hadoop	Java, C/C++	66.5	1643	http://hadoop.apache.org
Apache HBase	Java, Ruby, Thrift	14.1	362	http://hadoop.apache.org/hbase/
Condor	Java, C/C++	51.6	962	http://www.cs.wisc.edu/condor/
DSpace	Java	23.4	217	http://www.dspace.org
Ganglia	C	19.3	22	http://ganglia.info
GLIDE	Java	2	57	http://sunset.usc.edu/~softarch/GLIDE/
Globus 4.0 (GT 4.0)	Java, C/C++	2218.7	2522	http://www.globus.org
Grid Datafarm	Java, C	51.4	220	http://datafarm.apgrid.org/
Gridbus Broker	Java	30.5	566	http://www.gridbus.org/
Jcgrid	Java	6.7	150	http://jcgrid.sourceforge.net/
OODT	Java	14	320	http://oodt.jpl.nasa.gov
Pegasus	Java, C	79	659	http://pegasus.isi.edu
SciFlo	Python	18.5	129	http://sciflo.jpl.nasa.gov
iRODS	Java, C/C++	84.1	163	https://www.irods.org/
Sun Grid Engine	Java, C/C++	265.1	572	http://gridengine.sunsource.net/
Unicore	Java	571	3665	http://www.unicore.eu/
Wings	Java	8.8	97	http://www.isi.edu/ikcap/wings/

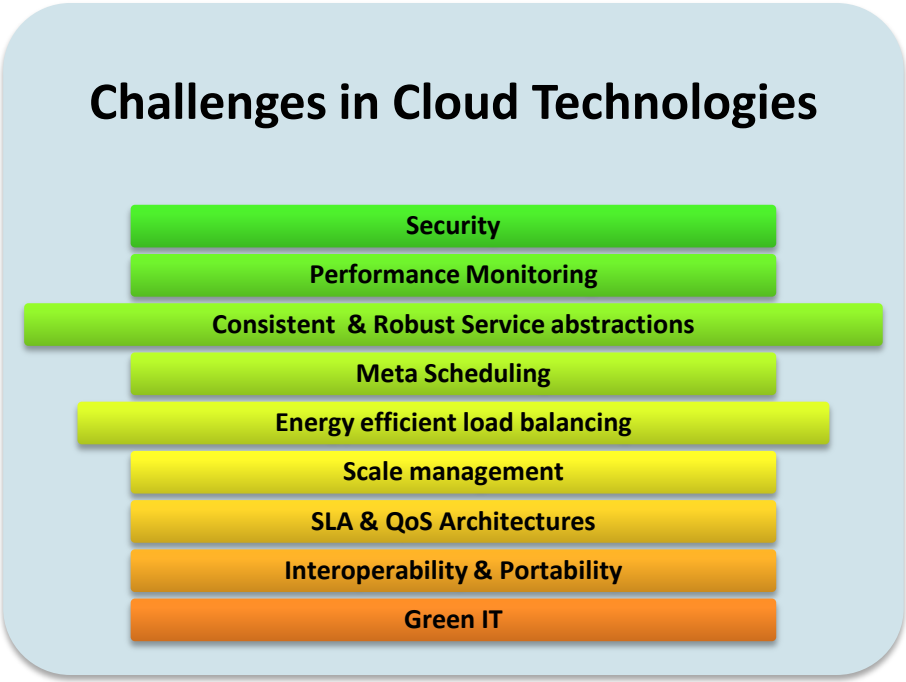
Reference: C. Mattmann, J. Garcia, I. Krka, D. Popescu and N. Medvidovic. The Anatomy and Physiology of the Grid Revisited. In *Proceedings of the Joint Working IEEE/IFIPConference on Software Architecture & European Conference on Software rchitecture,* pp. 285-288, Cambridge, UK, September 14-17, 2009.

DSSA for Grid Computing
---- Capturing Service Component Interactions



Migrating to the Cloud – The Issues and Challenges





The overall Migration Steps:

- Evaluate & Assess
- Pilot for the right level of migration
- Rearchitect, Redesign and Reimplement part or all of the components – The Hybrid design
- Leverage platform advantages
- Leverage Cloud Services Eco-Systems
- Test and Validate - including the pricing dimensions
- Refactor, Refine and Re-Iterate

Classifying Cloud Service Abstractions

Given

-- public cloud services operational opacity

-- fewer open source packages available for setting up one's own private 'cloud',

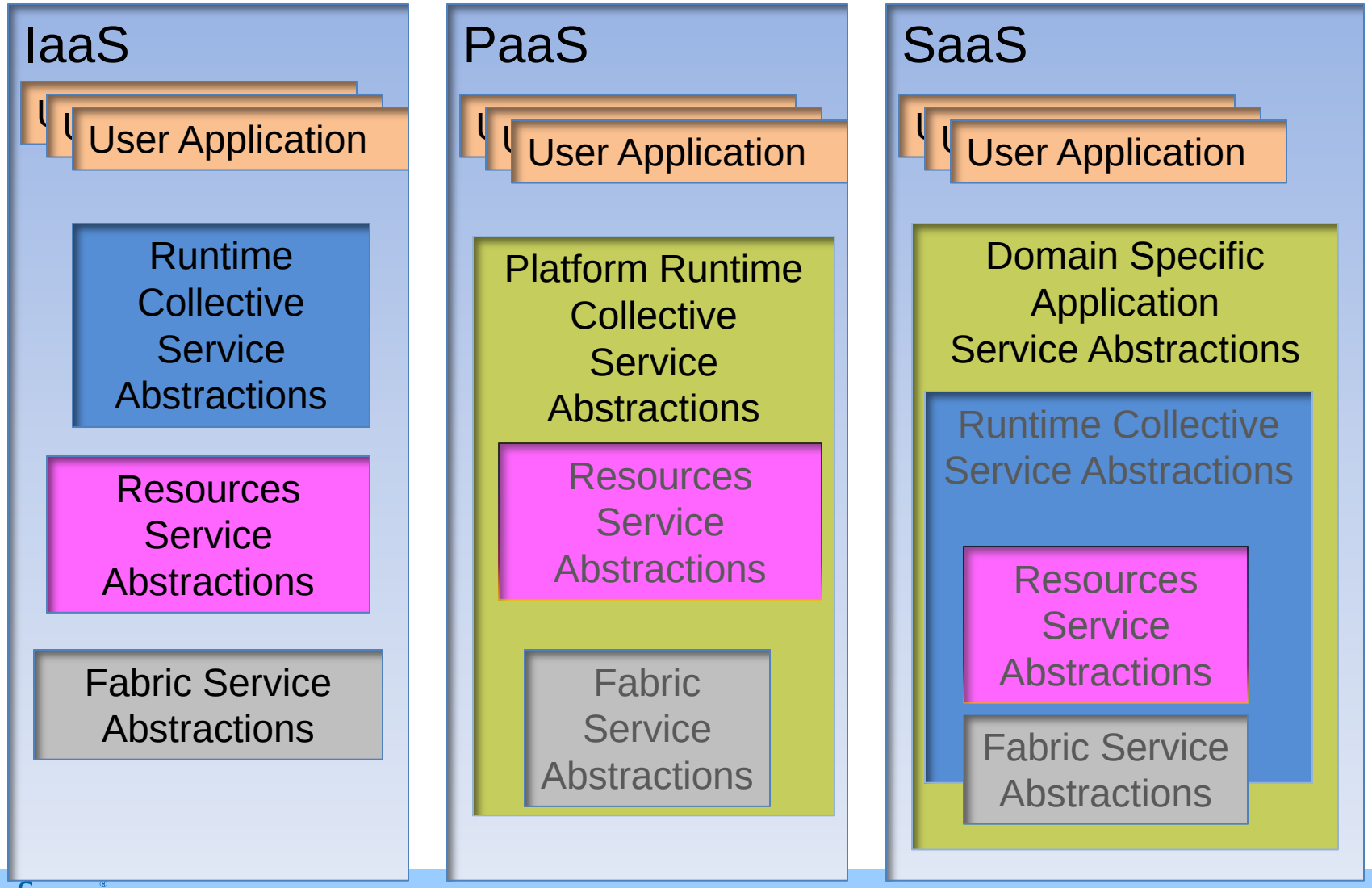
the key DSSA driven cloud service abstractions are:

- Domain Specific Application services abstraction
- Platform Runtime Collective services abstraction
- Runtime Collective services abstraction
- Resource services abstraction
- Fabric services abstraction

--- Encompasses Functionality and API from Grid DSSA

Work in progress : Study of several applications – structural, service component interactions, service abstraction interactions for each of the conventional cloud offerings: IaaS, PaaS and SaaS.

Domain Specific Software Architectures models for Cloud Service Abstractions



Leveraging the Cloud DSSA – The Best Practices for Hybrid Clouds

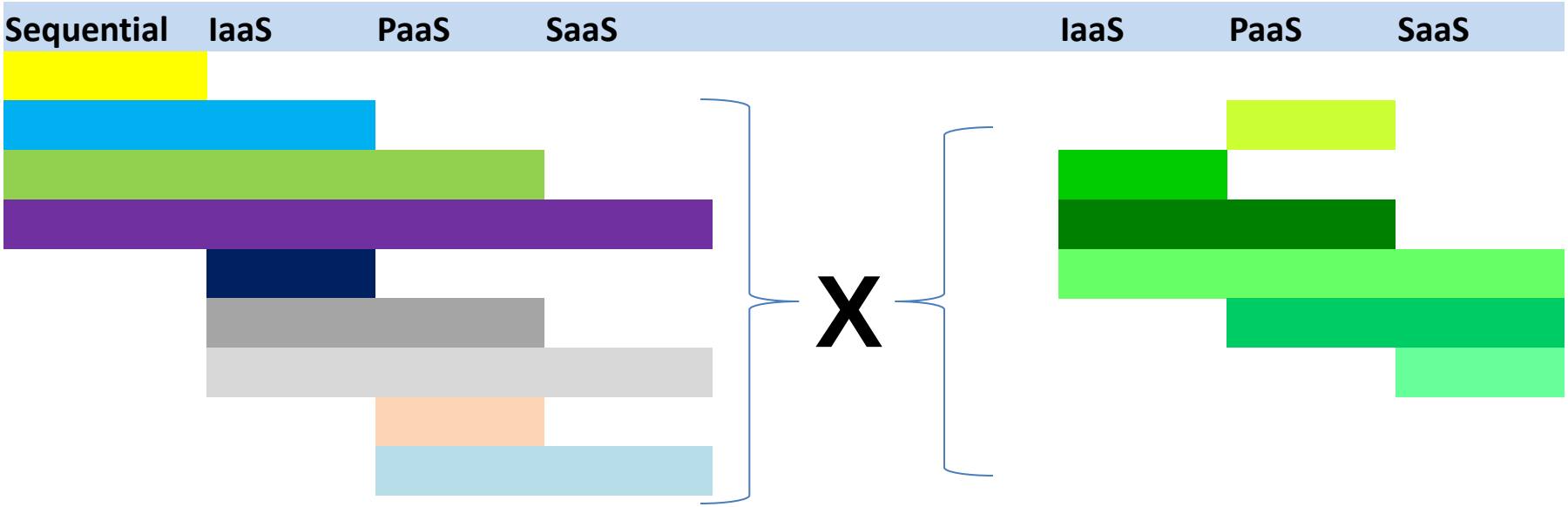
Key Challenges in taking an existing application into the Cloud Hybrid mode:

- Rearchitect, Redesign, Reimplement as necessary
- Issues: Interoperability, Portability, Security and Variable Pricing

Hybrid Cloud Migration and Deployments options for an application

Private Data Center & Private Cloud

Public Cloud Offerings



Thank you!!

Contact Information:
T S Mohan, PhD
Principal Researcher, E&R,
Infosys Technologies Ltd, 44 Hosur Road, Bangalore 560 100, INDIA
Email: subramanian_mohan@infosys.com