

Service Level Agreement in Service Centric System Engineering

Dr. Olivier Nano

Lead Software Design Engineer
European Microsoft Innovation Center
Email: onano@microsoft.com

What is a Service Level Agreement?

- Full flavoured definition:
 - A signed agreement between a consumer and a service provider describing what the service offers and under which conditions the service is offered
- Too often restricted to:
 - A set of Quality of Service (QoS) attached to a service

Why are SLAs helpful?

- Clearly define what the service delivers
- Set out price and penalties
- Identify competitive advantage
- Build collaborative relationships
- Clearly define obligations on both sides
- Define what is success and how it is measured
- Help to configure infrastructure to deliver the service

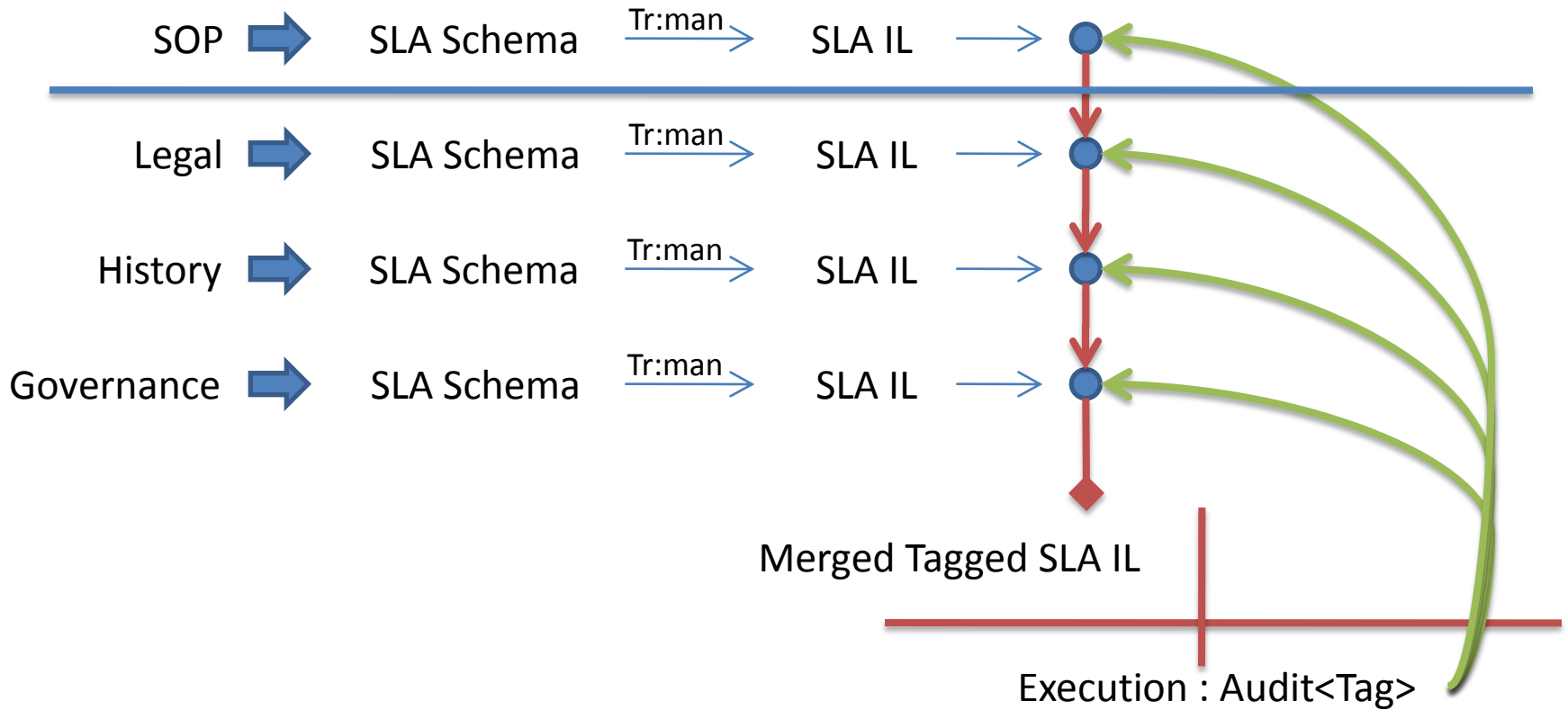
How to offer a service under a SLA?

- For each service or application offered under an SLA the provider will have to:
 - Decide how to deploy the service or application to meet the SLA guarantees
 - What are the resource requirements
 - Decide how much to charge for the service
 - Need economic viability
 - Create management policies to maintain QoS
 - Need automated management to reduce service times and costs
 - Respond to the customers request in a timely fashion
 - Needs to be fast
 - Maintain an efficient use of infrastructure
 - Avoid reservations – may lead to Economic Denial of Service

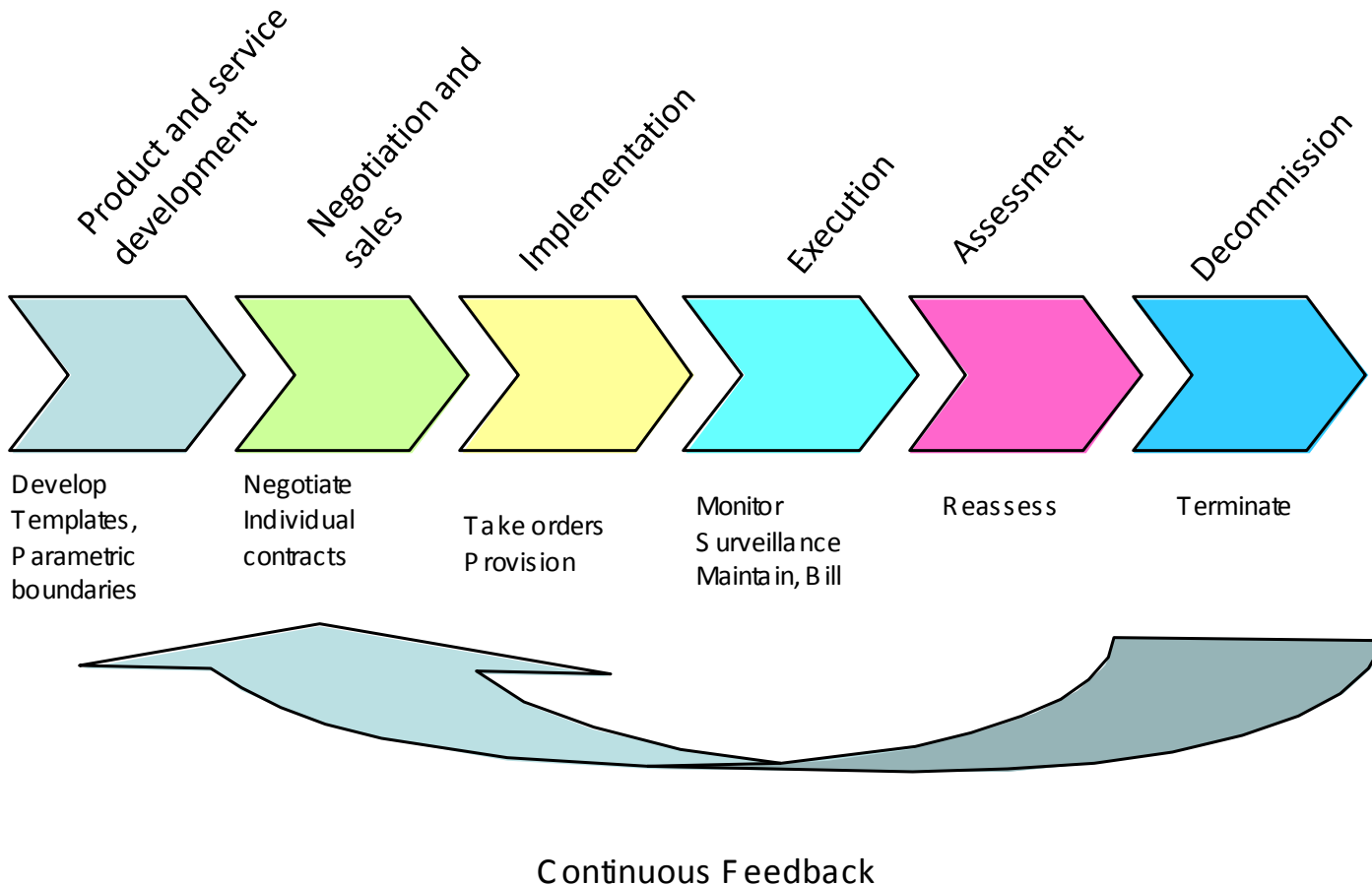
Rational when offering a SLA

- Problems
 - There is no easy way to predict code performance on various platforms
 - Initially a manual process to generate the management policies
 - Impact of technology on business goals not always obvious
 - Too many variables introduce complexity in the management space, exponential increase
- Service provider's solution
 - Rationalize Portfolio
 - Standard Well-Characterised Reusable Components
 - Discrete bands of service offerings with well described and characterized QoS
 - Reduce the number of variables guaranteed

Different views on a SLA



SLA Lifecycle



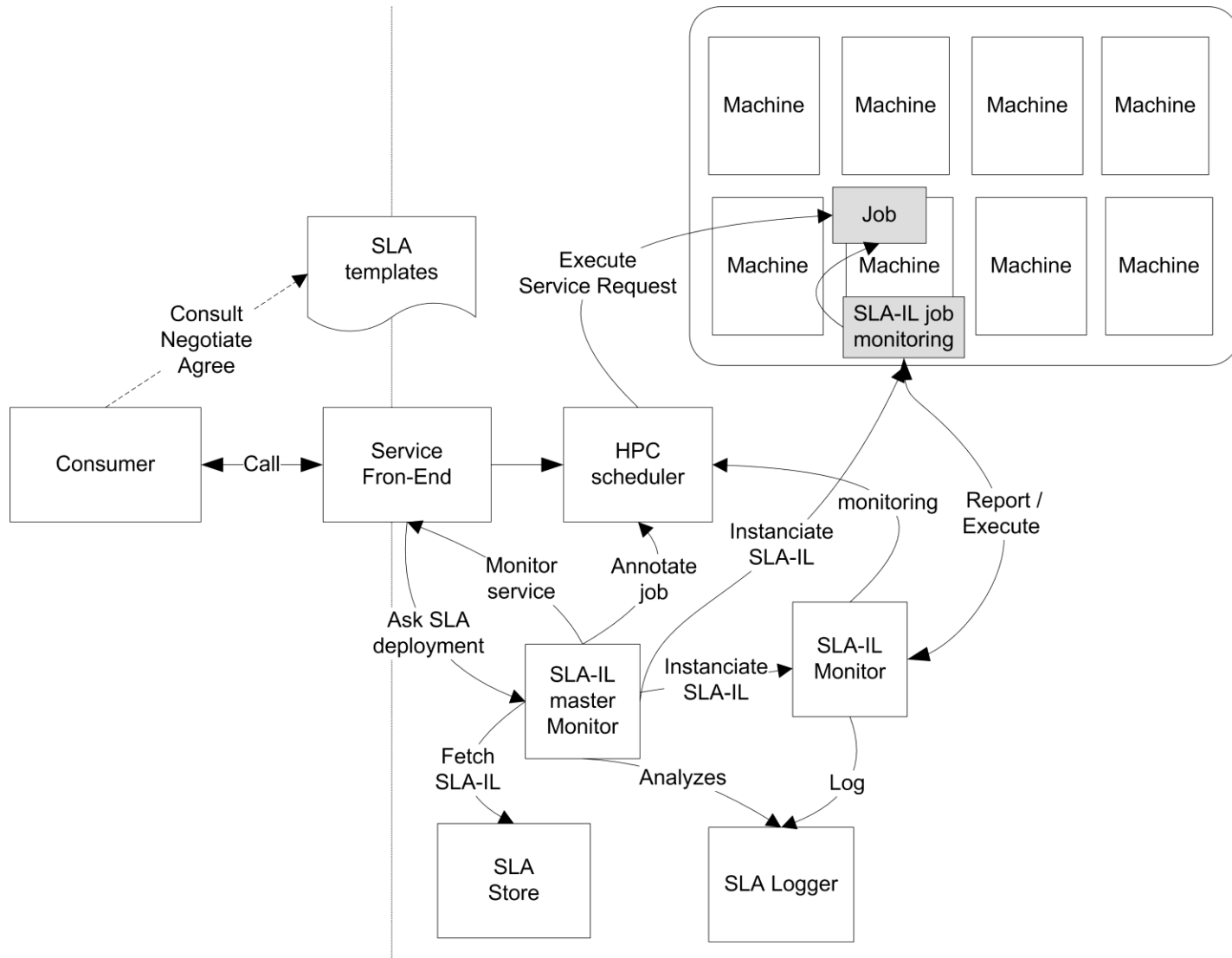
SLA in Service Centric Systems

- Service Level Languages
 - Plain English contract
 - WSLA/WS-Agreement (IBM, GGF)
 - Describes what metrics a service uses to evaluate level of service
 - Becoming very focused on GRID (GGF standardization)
 - Different requirements from commercial services
 - Cremona, IBM's middleware framework for WS-agreement
- WS Monitoring Products
 - AmberPoint
 - Service Level Management, Exception/Fault Tracking
 - Management, No consumer <-> provider communication
 - Customers: Microsoft, IBM, BT, Fujitsu, Orange/FT, etc...
 - Actional's Looking Glas
 - SOA Software's Service Manager

SeCSE Approach to SLA Language: A 3 Layers Architecture

- **Measurement:**
 - Collect & bundle information at defined points in time
 - 3 Types of information: infrastructure management, extracted from the messages, extracted from communication link
 - Triggered when a message is dispatch, when an error occurred or on timer
- **Computation:**
 - Aggregate information received from measurement and perform operations on information at defined point in time
- **Audit:**
 - Receive information from measurement and computation and check if it's in the range of the level objectives

Deploying SLA at runtime



Conclusion

- SLA is important to customers and key differentiator between providers
- SLA are generally bi-partite
- Business Level Objectives should be the focus
- Number of actors in the SLA lifecycle
- Need to support collaboration
- Management simplification through discrete set of offers