The SLA@SOI Project and Custon are joining forces in their ambition to make the cloud-based service economy a reality by providing a service level management platform which enables IT services to be handled as economic goods.
The changing paradigm

In the cloud all that matters for the business and consumer end-user is the qualitative end user experience.

Underneath the SLA there is a myriad of physical and virtual objects of which the performance has to be monitored and measured and controlled down to the bottom of the SLA stack.

Cloudcomputing will change IT contract and service level management between now and the coming ten years fundamentally.

PerformWorks

In this white paper Custon and the SLA@SOI Project will explain there respective activities and in particular there efforts in combining there research and technology. More specifically the SLA@SOI Project framework and the Custon PerformWorks SLA SaaS platform which is already in use within several global fortune companies.

History Custon

Custon was founded in 2005 with the objective to realize a customer led SLA service infrastructure by measuring, monitoring and managing the quality of the business service perception. Our believe is that the true value of services lies in the perception of the end-user and that was long misunderstood in the IT landscape. We knew that in order to engineer service excellence it was paramount to bridge between IT en business using feedback from the business stakeholders in all stages of the service life cycle process.

Adapting the new service stack cloud paradigm

Adapting the new cloud paradigm is not a question of if but when for most companies. Whether we speak about a private or a public cloud most organizations will face a Business IT service chain configured with more and more OEM cloud building blocks. These building blocks can be the infrastructure service provider, the application service provider or the business process service provider. In the cloud there will be numerous ways to build and deliver the service stack. It is clear that to be effective all service building blocks must interact fluently in order to support the business in her execution of the business plans and market development. There is a lot at stack because an underperforming SLA service stack can hurt business performance dramatically. This means that adapting insourcing, outsourcing or other cost effective service delivery models like SaaS can not be decided partially but must be decided holistically. Hence the service chain is as weak as her weakest service building block and must therefore be monitored and managed within the SLA metric.

Our Vision: A business-ready service-oriented infrastructure empowering the service economy in a flexible and dependable way.
The down site of sub optimizing mainly on cost drivers is losing control over the service delivery chain and ends up seeing the total cost rise and quality going down. At the same time we see that demand site orchestration of the service stack becomes harder and harder because outsourcing as the new mantra in most cases includes outsourcing of the information management as well.

Since the beginning of the crisis business is blaming IT for focusing only on the cost drivers and neglecting value and risk drivers. On the contrary business is interested in flexible and agile running operations to adapt to market changes. In there discussion the business case is key. Unfortunately the orchestration of demand and supply operations depends on the goodwill of external parties or is outsourced completely to vendors. In the Netherlands alone this wave of sub optimization in the service delivery value chain has let to a loss of billions of euro’s.

The PerformWorks product vision

The solution to meet the challenge of managing the diffuse service stack is in the collection of feedback across the whole service stack. This means from functional qualitative to technical quantitative. This approach guarantees a good insight in how the different service layers are executed and interrelate to each other. Custon envisioned this in her so called OSISS management model describing the Open System Interconnect Service Stack model in which SLA causality in the service stack is monitored and managed. This vision shaped the fundamentals of the SaaS platform PerformWorks addressing the need for data collection in the total Business IT Value chain analyzing the service stack weaknesses expressed by SLA non performance. Study learned that these weaknesses exists because there is no historical interfacing between the different IT frameworks.

As we analyze the frameworks within the IT landscape like BiSL, ASL, ITIL, CobIT we notice that there are no framework API’s or even a concept addressing the interconnection between the frameworks. This subject becomes nevertheless very important considering the fact that managing the Business IT service stack means managing frameworks alongside the service stacks. Notwithstanding the fact that the individual frameworks are more than meaningful on there own from a holistic view however an exchange platform is missing for connecting these frameworks together with there inherent processes and SLA’s. The OSISS management model Custon is using distributes data point collectors throughout the OSISS stakeholder domain i.e. service providers, service brokers, application providers, infrastructure providers and business units. Data is collected, verified and transformed into SLA related KPI’s representing the quality of service.

Because a service stack is often composed out of different service building blocks (outsourced, in sourced or bought as SaaS or ASP platform) this means that the stack must be SLA chain aware. Therefore SLA causality must be part of the monitoring and measuring process. SLA performance transparency guarantees that defects in a lower service layer are detected and solved before it influences the outcome of the upper service layer. The PerformWorks platform keeps a balance between qualitative and quantitative data collection.

For example when service delivery is interested in business functional SLA feedback a large library is at disposal in which qualitative application aspects of SAP or Oracle are ready for use to measure business perception in conjunction with infrastructure performance.

The Custon program code

The Custon user interface is written in Ajax code and implemented on MS .net technology. The GUI logic is based on numerous feedback from our customers representing blue chip national and international organizations. From a conceptual angle there are four modules in the platform i.e. the SLA content management module, the data collection module, the statistic module and the presentation module. The program code was chosen to adapt to cloud computing and be flexible towards market changes. Cloud customers are not interested in technology but functionality. We are confident that our technology can assure this cloud demand in the future.

A lot of energy was invested in data security because this is a cloud imperative.
Why did Custon choose for object programming? The emphasis within PerformWorks is on data rather than procedure. Secondly data is hidden and can’t be accessed by external functions. Programs are divided into what are known as objects. Objects may communicate with each other through methods. What do we think are the benefits of Object Oriented Programming? Through Inheritance, we can eliminate redundant code and extend the use of existing classes. We can build programs from the standard Working modules, rather than writing the code from scratch. This leads to savings of development time and gives higher productivity. The principle of data hiding helps from invades by code in other parts of the program. Easy to partition work in our projects based on Objects also benefits. Object oriented systems can be easily updated from small to large systems. Software complexity can be easily managed. Custon used the following model to develop the PerformWorks application, which is roughly similar the MS dot.net framework model.

Custon principal design features

In our developing we strive for quality and operate according to principal design features, which are the following.

Interoperability. Because interaction between new and older applications is commonly required, the .NET Framework provides means to access functionality that is implemented in programs that execute outside the .NET environment. Access to COM components is provided in the System.Runtime.InteropServices and System.EnterpriseServices namespaces of the framework; access to other functionality is provided using the P/Invoke feature.

Common Runtime Engine. The Common Language Runtime (CLR) is the virtual machine component of the .NET framework. The Custon .NET programs execute under the supervision of the CLR, guaranteeing certain properties and behaviors in the areas of memory management, security, and exception handling.

Language Independence. The .NET Framework introduces a Common Type System orCTS. The CTS specification defines all possible datatypes and programming constructs supported by the CLR and how they may or may not interact with each other conforming to the Common Language Infrastructure (CLI) specification. Because of this feature, the Custon .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET Language.

Base Class Library. The Base Class Library (BCL), part of the Framework Class Library (FCL), is a library of functionality available to all languages using the .NET Framework. The BCL provides classes which encapsulate a number of common functions, including file reading and writing, graphic rendering, database interaction, XML document manipulation and so on. In Custon terms this means speed of execution.

Simplified Deployment. The Custon .NET framework includes design features and tools that help manage the installation of computer software to ensure that it does not interfere with previously installed software, and that it conforms to security requirements.

Security. The design is meant to address some of the vulnerabilities, such as buffer overflows, that have been exploited by malicious software. Additionally, .NET provides a common security model for all applications.

Portability. The design of the .NET Framework allows it to theoretically be platform agnostic, and thus cross-platform compatible. That is, a program written to use the framework should run without change on any type of system for which the framework is implemented.

ASP.NET Architecture

In her roadmap investments Custon is anticipating the cloud by developing a multi SLA focused service stack monitoring, measuring and improvement SaaS platform. The R&D department of Custon is constantly researching the market for roadmap opportunities. The DNA of the roadmap is based on the concept that service structure follows market strategy and therefore services must be business driven. The simple fact that end customers will demand quality of service from the business will in return mean that business will demand service quality from IT. IT in return will demand high quality service levels from her vendors. The above quality of service levels can only be established when cascading SLA’s are constantly monitored and controlled.
The technology roadmap of Custon is aimed at:

- further integration of SLA management meaning implementing SLA templates using dynamic SLA KPI fields and a common service level language.
- closer integration and performance visualization of the whole service stack
- implementing interfaces in order to interact with virtualized infrastructures
- developing business logic and algorithms for predictable and adjustable services.

Our strive for product perfection brought us on the path of the SLA@SOI Project.

After due consideration it seemed logical to share our combined visions and project scope because the SLA@SOI Project envisions the same objectives in there roadmap, implying:

- an automated e-SLA contracting framework,
- systematic grounding of SLAs from the business level down to the infrastructure,
- exploitation of virtualization technologies at infrastructure level for SLA enforcement, and
- advanced engineering methodologies for creation of predictable and manageable services.

**Combined vision Custon and the SLA@SOI Project.**

SLA@SOI is an FP7 ICT 2007 Call 1 Integrated Project addressing Objective 1.2 SSAI 'Service and Software Architectures, Infrastructures and Engineering'. Launched on June 2nd, 2008, the SLA@SOI Project is committed to research, engineer and demonstrate technologies that can embed SLA-aware infrastructures into the service economy. SLA@SOI targets SLA-driven management, and monitoring the life-cycle of services such as software and infrastructure services.

The SLA@SOI Project envisions in correspondence with Custon a business-ready service-oriented infrastructure empowering the service economy in a flexible and dependable way where business-readiness requires the following three major characteristics.

**Predictability and Dependability:** The quality characteristics of services can be predicted and enforced at run time.

**Transparent SLA management:** Service level agreements (SLAs) defining the exact conditions under which services are provided/consumed can be transparently managed across the whole business and IT stack.

**Automation:** The whole process of negotiating SLAs and provisioning, delivery and monitoring of services will be automated allowing for highly dynamic and scalable service consumption.
The ongoing transformation of a product-oriented economy towards a service-oriented economy has come to a critical point. IT-supported service provisioning is of major relevance in all industries and domains. However, the nature of these setups is typically quite static because it requires significant effort to create service offers, to negotiate provisioning details with customers and to manage and control provided service.

The SLA@SOI research will provide a major milestone for the further evolution towards a service-oriented economy, where IT-based services can be flexibly traded as economic goods, i.e. under well defined and dependable conditions and with clearly associated costs. Eventually, this will allow for dynamic value networks that can be flexibly instantiated thus driving innovation and competitiveness.

The conceptual approach of SLA@SOI is to define a holistic view for the management of service level agreements (SLAs) and to implement an SLA management framework that can be easily integrated into a service-oriented infrastructure (SOI).

Cloud computing will change IT contract and service level management between now and the coming ten years fundamentally. In the cloud all that matters for the business and consumer end-user is the qualitative end user experience. Underneath the SLA there is a myriad of physical and virtual objects of which the performance has to be monitored and measured and controlled down to the bottom of the SLA stack.

Service live cycle management

Service live cycle management will have a totally different meaning in the cloud (private or public). In the cloud there are three different pure service providers all having a fundamental role in the cloud service delivery. We define the business service provider (broker), software service provider and the infrastructure service provider. After completing a business assessment the customer who inducted a specific service demand is addressed to the service provider. The business service provider in return is starting a terms & conditions SLA negotiation process which after successful completion will be monitored and arbitrated in case of violation.

The figure above gives a simplified overview on how such a systematic SLA management process may look like. As today’s business systems typically consist of complex layered systems, user-level SLAs cannot be directly mapped onto the physical infrastructure. Services might be composed of other more fundamental services that could be also provided by external parties. Consequently, a stepwise mapping of higher-level SLA requirements onto lower levels and the aggregation of lower level capabilities to higher levels is crucial for grounding user-level SLAs to the infrastructure. The service provider must guarantee that the agreed terms in the SLA are fulfilled and does this by a real time checking of the necessary resources during the SLA negotiation phase.

The pure software provider in return is responsible for delivering the right software modules at the right time against the agreed upon conditions. This urges the software provider to take care for the right design and time repository, a well balanced software landscape and a strong organized software deployment and management.
The pure play infrastructure provider provisions the SLA according to the negotiated terms through virtual resource allocation and management. This is orchestrated on the infrastructure landscape by allocating and virtualization of psychical resources which are controlled by actors and sensors. The virtual infrastructure is managed by IT operation rules stored in the policy repository.

Integrated SLA chain management is in many ways the new paradigm because cloud computing whether it is private or public is approaching fast and lacks a holistic service performance management grounding. Cloud computing is the new mantra in achieving flexibility, adaptability and scalability. At the same time it is lowering cost by using the economies of scale of large cloud service providers and introducing on demand pay per use technologies in infrastructure, middleware and primary applications. The only thing that is missing is an integrated SLA chain management process. That is were Custon and the SLA@SOI Project bundle forces. In the picture below a service chain or stack implementation is visualized in which the business has his own commercial SLA towards the market and fully depends on the SLA driven Service Oriented Infrastructure build in the cloud. The Service Oriented Infrastructure (SOI) represents an architectural shift for building business applications based on loosely coupled services. In a multi-layered SOI environment the exact conditions under which services are to be delivered can be formally specified by Service Level Agreements (SLAs). However, typical SLAs are just specified at the top-level and do not allow service providers to manage their IT stack accordingly as they have no insight on how top-level SLAs translate to metrics or parameters at the various layers of the IT stack. This is one of the most compelling reasons that the zone of service compliance is shrinking over time with an alarming speed. The only way to overcome this jeopardy is the introduction of integrated SLA chain management. How will Custon and the SLA@SOI Project together fill in this challenge?

The answer is that the SLA@SOI Project recently has proposed a conceptual framework for the precise definition and classification of SLA translations in SOI. Custon has studied this framework and concluded that it fits into the Custon roadmap specifications.

In order to optimize the whole SLA value chain it is obligatory that service providers are acting as one SLA mechanism up and down the value chain without redundancy and by using all the information available. Only than predictive modelling can be implemented creating an unprecedented user experience.
C2C (Configuration to Configuration): this type of translation mostly relates to the dependencies within a layer or between layers. Such dependency graphs are useful in configuration management and problem diagnosis.

M2C (Metric to Configuration): this type of translation translates higher-level objectives to lower-level system parameters. It can also be referred to as “top-down” translation or SLA decomposition. It is useful for sizing and capacity planning, mostly at design time.

C2M (Configuration to Metric): this type of translation predicts higher-level objectives from lower-level system parameters. It can also be referred to as “bottom-up” translation or performance prediction. It is useful both for what-if analysis at design time and predictive management at run time.

M2M (Metric to Metric): this type of translation correlates a high-level metric with lower-level metrics. The translation can go both directions, namely decomposition or prediction, depending on the usage scenario. It is useful for forecasting and problem diagnosis at run time.

Important message form the above picture is the fact that for example SAP business services will be composed from a myriad of resources, components, services and processes all interconnected through observable metrics and configurable parameters.

In order to support business units from the cloud with critical services like HR and CRM all metrics have to be managed in a single monitor view with 90% of the service chain self-propelling meaning adjusting resources at real-time on prediction and usage scenario scripting.

Infrastructure SLA and Resource Model

In the end SLA’s within SLA@SOI are conceptually comprised of a Set of terms: Terms in an SLA are the attributes that define the eventual representation of an instantiated resource. These attributes within the SLA@SOI framework are broadly split into two categories, functional and non-functional and are in principle immutable. We mention a Set of SLOs: SLOs are explicit and possibly implicit attributes that are monitored for changes in their value and are associated at runtime properties of the resource. Management Rules: Where SLA terms and/or SLOs change above or below a particular explicit or calculated values then a set of associated actions are executed. The SLA@SOI translation framework together with the infrastructure SLA and Resource model are fundamental for the new service economy. Custon has already build a best practice SLA manage SOI platform which is performing at blue chip companies in the Netherlands. Together with the SLA@SOI working group Custon will further develop her roadmap towards a fully manageable SLA@SOI platform.
The research project SLA@SOI aims at providing a major milestone for the further evolution towards a service-oriented economy, where IT-based services can be flexibly traded as economic goods, i.e. under well defined and dependable conditions and with clearly associated costs. Eventually, this will allow for dynamic value networks that can be flexibly instantiated thus driving innovation and competitiveness.

The next step

The direction is clear. It’s not if but when cloud computing will be pervasive. There are huge advantages why to embrace cloud computing. Value and cost drivers like pay per use economics, time to market factors, speed of execution, use when needed are just some of the arguments. Off course this process will be slowed down if we don’t address the risk drivers. Custon and the SLA@SOI Project will share there knowledge in order to address issues like IT to strategic business service transformation, integrated SLA service oriented architecture management, efficient SLA standardized content management, interaction between metric and parameters on behalf of predictive SLA provisioning at runtime and SLA service oriented infrastructure repeatable solutions conform the mantra build ones and use many.

In the coming next months Custon will start a initiative which is called SLA@SOI chapter the Netherlands. This chapter is supported by the SLA@SOI Project. In this chapter we will invite industry leaders in the Netherlands to actively participate in knowledge sharing and by doing this inherently anticipate on this new paradigm called SLA@SOI. We will ask members of the SLA@SOI chapter the Netherlands to test our software within there organizations and by doing this get familiar with this new phenomenon which will affect business all over the world.

Organizations that form part of the SLA@SOI Project are SAP, Telefonica, Intel, Technical University Dortmund, City University London and others.