Universal Integration of the Internet of Things through an IPv6-based Service Oriented Architecture enabling heterogeneous components interoperability

Grant agreement for: Collaborative project
Grant agreement no.: 288445

Start date of project: October 1st, 2011 (36 months duration)

**Deliverable D7.4**
*Innovative Business Processes Test and Validation Report*

<table>
<thead>
<tr>
<th>Contract Due Date</th>
<th>31/07/2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission Date</td>
<td>20/08/2014</td>
</tr>
<tr>
<td>Version</td>
<td>v1.0</td>
</tr>
<tr>
<td>Responsible Partner</td>
<td>RMP</td>
</tr>
<tr>
<td>Author List</td>
<td>Lou Fedon, Ian Thomas, Sébastien Gaïde</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>PU</td>
</tr>
<tr>
<td>Keywords</td>
<td>Internet of Things, Business Process</td>
</tr>
</tbody>
</table>

Project Coordinator: Mandat International (MI)
Sébastien Ziegler  sziegler@mandint.org

This project has received funding from the European Union’s Seventh Framework Programme for research, technological development and demonstration under grant agreement no 288445.
Deliverable Abstract

The purpose of this deliverable is to document the activities and outcomes of Task T7.3 “Demonstrating new business processes and combining the various IoT6 components”. It describes how a business process management tool (namely the RunMyProcess Platform), that has been enhanced with IoT integration capabilities, can be used to design and run novel business processes that link the physical world to the digital one and vice-versa.
Table of Contents

Deliverable Abstract .................................................................................................................... 2
List of Figures................................................................................................................................. 4
List of Acronyms ............................................................................................................................ 6
1. Introduction ................................................................................................................................. 7
   1.1. Purpose and Scope of the document ....................................................................................... 7
   1.2. Task T7.3 .............................................................................................................................. 7
   1.3. Structure of the Document .................................................................................................... 7
2. RunMyProcess Features .............................................................................................................. 8
   2.1. Business Processes in RunMyProcess .................................................................................. 8
   2.1.1. Web Interfaces .................................................................................................................. 10
   2.1.2. Process ............................................................................................................................. 13
   2.1.3. Process and Web forms interaction: Launch Forms & Manual Tasks ......................... 17
   2.1.4. Other modes to trigger a process: Scheduling and Email ........................................... 21
   2.2. Integration in RunMyProcess .............................................................................................. 23
   2.2.1. Legacy Web Service connectors ...................................................................................... 23
   2.2.1.1. Connectors pre-developed library .............................................................................. 25
   2.2.2. CoAP connector .............................................................................................................. 26
   2.2.3. Composite API ............................................................................................................... 27
   2.2.4. CoAP Proxy .................................................................................................................... 28
3. Web of Everything - A Platform for Composite Business ...................................................... 29
   3.1. The Ever Expanding Web .................................................................................................... 29
   3.2. Platforms for Composite Business Ecosystems .................................................................. 30
   3.3. Extending to the Web of Things .......................................................................................... 31
   3.4. Benefits of a Cloud Mediated Approach to the Web of Everything .................................. 33
   3.4.1. Simplification and Externalization of Function ................................................................. 33
   3.4.2. Composition and Abstraction ......................................................................................... 34
   3.4.3. Resource Management .................................................................................................. 34
   3.4.4. Service Convergence ...................................................................................................... 34
   3.4.5. Unified Discovery, Subscription and Monetization .......................................................... 35
   3.4.6. Insight and Analytics ....................................................................................................... 35
   3.5. Broader Implications of Web Convergence and Composite Business Models ............... 36
4. Innovative Business Processes .................................................................................................. 37
   4.1. Part Replacement Use Case ................................................................................................. 37
   4.2. Smart meeting room demo (how we combine RMP features to link the real world with the web) ......................................................................................................................... 40
   4.3. I Heart Studios Use Case ..................................................................................................... 44
   4.4. Potential scenarios .............................................................................................................. 49
   4.4.1. Supply shortage ............................................................................................................... 49
   4.4.2. RFID Kanban (Integration with Trello and Bugzilla) .................................................... 49
5. Conclusions ................................................................................................................................. 51
References .......................................................................................................................................... 52
List of Figures

Figure 1 : Business Process Graphical Representation ................................. 8
Figure 2: RunMyProcess Flex IDE Client .................................................. 9
Figure 3: Web Interface Design ................................................................... 9
Figure 4: Process Design ............................................................................. 10
Figure 5 : A designed Web Interface ............................................................... 11
Figure 6: Corresponding rendered Web Interface ......................................... 11
Figure 7: Composite API Design .................................................................. 12
Figure 8: Composite API call configuration in a Web Interface ...................... 13
Figure 9: Process Design Example ................................................................. 14
Figure 10: Process with a Loopback Example ................................................. 14
Figure 11: Assignment of a variable ................................................................. 15
Figure 12: A launched process execution path .............................................. 15
Figure 13: A launched process input variables .............................................. 16
Figure 14: A launched process computed variables .................................... 16
Figure 15: A launched process internal variables ...................................... 16
Figure 16: Launch Process Button Design .................................................... 17
Figure 17: Web Interface with a Process Launch Button ............................... 18
Figure 18: Process Launch Confirmation ...................................................... 18
Figure 19: Input variables passed from the Web Interface ............................. 19
Figure 20: Screen Configuration ................................................................. 19
Figure 21: Manual Task Configuration .......................................................... 20
Figure 22: Launched Process waiting for a manual task validation .............. 20
Figure 23: Manual Task .............................................................................. 21
Figure 24: Email Trigger Configuration ........................................................ 22
Figure 25: Schedule Trigger Configuration .................................................. 22
Figure 26: Web Service Call Configuration .................................................. 23
Figure 27: Provider Configuration ................................................................. 24
Figure 28: Connector Configuration ............................................................... 25
Figure 29: Public connector library .............................................................. 25
Figure 30: CoAP connector ......................................................................... 26
Figure 31: CoAP connector configured in a process step ............................ 26
Figure 32: Composite API Example ............................................................. 27
Figure 33: Composite Business Platform ...................................................... 30
Figure 34: Web of Everything ..................................................................... 32
Figure 35: A Platform for Convergence ....................................................... 32
Figure 36: Benefits of Convergence .............................................................. 33
D7.4 Innovative Business Process Test and Validation Report

Figure 37: Value Chain Reconfiguration ............................................................. 36
Figure 38: Part Replacement Use Case – Alert Notification API ......................... 37
Figure 39: Part Replacement Use Case – Alert Notification .................................. 37
Figure 40: Part Replacement Use Case – Notification phone and email configuration ........................................................................................................... 38
Figure 41: Part Replacement Use Case – Alert has been marked as resolved........ 38
Figure 42: Part Replacement Use Case – Marking an alert as a false positive ....... 38
Figure 43: Part Replacement Use Case – Creating a replacement request ........... 39
Figure 44: Part Replacement Use Case – Retrieving the order status ................... 39
Figure 45: Part Replacement Use Case – Closing the part replacement order ....... 39
Figure 46: SmartMeeting Use Case - Meeting Reservation Web Form ............... 40
Figure 47: SmartMeeting Use Case - Meeting Google Calendar Event ............... 41
Figure 48: SmartMeeting Use Case - Scheduled Meeting Preparation ............... 41
Figure 49: SmartMeeting Use Case – Authorization API .................................... 42
Figure 50: SmartMeeting Use Case – Scheduled End of the Meeting .................. 42
Figure 51: SmartMeeting Use Case – Meetings Follow-up database ................. 42
Figure 52: SmartMeeting Use Case – Participants Follow-up database .......... 43
Figure 53: SmartMeeting Use Case – Participants Attendance Follow-up database 43
Figure 54: Boxes Check-in Web Form ................................................................. 44
Figure 55: Boxes Check-in Barcode Scan ............................................................ 45
Figure 56: Rail Creation Web Form ................................................................... 45
Figure 57: Rail Creation Barcode Scan............................................................... 46
Figure 58: Rail physical Follow-up Sheet ............................................................ 46
Figure 59: Photo Shoot Web Form .................................................................... 47
Figure 60: I Heart Studios Boxes Check-in / Check-out process ....................... 48
Figure 61: I Heart Studios Rail shooting process ............................................. 48
List of Acronyms

API: Application Programming Interface
BPMN: Business Process Model and Notation
CAPI: Composite API
CoAP: Constrained Application Protocol
CSS: Cascading Style Sheet
ERP: Enterprise Resource Planning
GWT: Google Web Toolkit
HTML: Hypertext Markup Language
HTTP: Hypertext Transfer Protocol
IDE: Integrated Development Environment
IoT: Internet of Things
IPv6: Internet Protocol version 6
PaaS: Platform as a Service
SaaS: Software as a Service
SOAP: Simple Object Access Protocol
WSN: Wireless Sensor Network
1. Introduction

1.1. Purpose and Scope of the document

The IoT6 project aims at researching and exploiting the potential of IPv6 to develop a service oriented architecture overcoming the current Internet of Things fragmentation.

The purpose of this deliverable is to document the activities and outcomes of Task T7.3 “Demonstrating new business processes and combining the various IoT6 components”. It describes how a business process management tool (namely the RunMyProcess Platform), that has been enhanced with IoT integration capabilities, can be used to design and run novel business processes that link the physical world to the digital one and vice-versa.

1.2. Task T7.3

The goal of Task T7.3 is to demonstrate an innovative application scenario combining IoT6 with a mainstream ERP (Enterprise Resource Planning) or Business Process Management program accessible through Software as a Service (SaaS) provided through the Cloud.

The demonstration involves the various components of the IoT6 architecture, including mobile network communications, STIS, business process applications and heterogeneous devices.

This Task demonstrates the ubiquity of the IoT6 architecture by implementing an integrated use case scenario that involves all the research partners’ premises and locations interacting with each other through a distributed testbed.

1.3. Structure of the Document

The document first describes the RunMyProcess platform and details its main features; the business process design and execution engine and its integration capabilities. Detailing these features serves as a common ground to imagine other innovative processes than the ones that have been already developed and implemented.

We then present in Section 3 “Web of Everything – A platform for Composite Business”, a vision of what can be done using this integration between IoT and the Cloud Computing world. The Task T7.3 can be seen as a concrete step to make this vision real.

Section 4 “Innovative Business Process” details the two main business processes implemented during the course of the IoT6 project, as well as a live business process developed for a private RunMyProcess customer that proves the interest of private companies for integrating IoT into business SaaS applications.

Lastly, we suggest several other use cases that could be directly implemented. The feasibility of these scenarios are demonstrated in Section 2 “RunMyProcess Features”.

2. RunMyProcess Features

One of the IoT6 project goals, among others, has been to enable the integration of connected objects and classical web resources through business process management tools. This has been realized by deploying the IoT6 stack on the RunMyProcess platform, a PaaS that enables the design and the execution of business applications and processes.

The business processes that have been designed rely heavily on the available features of the RunMyProcess platform and in particular on the interaction between its two main features:

- The design and execution of interconnected processes and web forms, and
- The design of integration connectors that enable the exploitation from RunMyProcess applications of a third-party exposed Web Services.

To understand how it has been possible to develop the business processes that will be detailed in Section Innovative Business Processes and to imagine other use cases, we present the following two main features:

- Business Processes in RunMyProcess
- Integration in RunMyProcess

2.1. Business Processes in RunMyProcess

A business process is a collection of activities or tasks that detail a certain high-level task. It can often be visualized with a flowchart as a sequence of activities with interleaving decision points.

In RunMyProcess, a business process represents a combination of processes (a graph of tasks) and web interfaces that will define the data flowing through the execution of the process as well as their graphical representation.

The process part follows the Business Process Model and Notation (BPMN) - a normalized graphical representation.

![Figure 1: Business Process Graphical Representation](image)

The RunMyProcess platform enables the design and execution of business processes and aims at making the design and the execution of business process accessible to the largest number.

Though the designable web forms and process should be kept highly configurable, the goal is to make the development, test and release of a business process as easy and quick as possible. The designer should only focus on what is truly relevant for
his/her application.

Moreover, the platform is used to manage the execution of business processes and manage all the related operational aspects such as monitoring and versions upgrade.

The RunMyProcess IDE (Integrated Development Environment) client is part of the platform and is dedicated to the design, development and configuration of processes and web forms. Both of which can be designed graphically.
In the following paragraphs, the process and web forms as defined in RunMyProcess is presented more thoroughly as well as the link between these two concepts and some related features.

2.1.1. Web Interfaces

Web interfaces consist in a set of elements called widgets. A widget can be among other things:

- Text input
- Number input
- Date input (associated to a date picker to ease the date selection)
- Some HTML code
- File uploader
- Section
- Tabs
- Dropdown list
- Checkbox
- Radio button
All these widgets are positioned graphically into the web interface. Once a widget has been added to a web interface, the designer can configure it. Depending on the type of widget, the configuration varies.

Once the web interface has been designed and saved, it can be displayed and tested. The rendering engine RunMyProcess web interfaces has been developed on top of the GWT framework. The web interface is rendered in HTML by the GWT framework.
D7.4 Innovative Business Process Test and Validation Report

In addition to their graphical design, the rendering of web interfaces can be modified by CSS (Cascading Style Sheet) configuration.

Their behavior can be enriched by adding JavaScript’s scripts that can be triggered at the opening of the web interfaces or when the value of a field is modified.

Another powerful option is to trigger the execution of a Web Service or a combination of Web Services directly from the web interface. The result of the call is retrieved in the context of the interface, which means that web interfaces are able to retrieve data from third-party providers and display it in order to enrich their behavior. This feature is called API Listener and has been described in deliverable D6.5 “Business Process Management Tools and Cloud Computing Applications Integration Report”.

To summarize, a Composite API can be designed graphically like a process but is executed synchronously and can be simply integrated in the RunMyProcess Web interfaces.

Figure 7: Composite API Design
In Figure 8: Composite API call configuration in a Web Interface, the Composite API is configured to be triggered each time the web interface is opened. If the Composite API call returns an HTTP error code, the failed script gets executed. On the other hand, if the call returns an HTTP success code, the completed script gets executed.

It is important to note that in RunMyProcess, a web interface can contain several screens. A screen is a particular view of the web interface. In a screen, a widget present in the web interface can be configured to be invisible or read-only. This very important concept will be explained in more details in Section Process and Web forms interaction: Launch Forms & Manual Tasks.

2.1.2. Process

As previously stated, a process is a sequence of activities with interleaving decision points called gateways. Activities and gateways are added to a process also by using a graphical designer. They can then be linked to specify the way their sequence should be executed.

Activities can be either:
- Manual tasks (web forms to be approved for the process to resume)
- Sending Emails
- Web Service calls
- Timers (a defined pause in the execution of the process)
- Sub-processes launch
- Computations of a long-running script

Gateways can either be:
D7.4 Innovative Business Process Test and Validation Report

- Split XOR (depending on a condition the process will proceed in a different branch of the graph)
- Split AND (the process will proceed in 2 different branches in parallel)
- Join XOR (used to close the potential 2 branches)
- Join AND (used to close a parallel execution)

Figure 9: Process Design Example

Combining a Join XOR and a Split XOR can enable the process to loopback on a defined activity or even a part of the graph.

Figure 10: Process with a Loopback Example

In addition to executing activities sequentially or in a parallel way, one can add computation in each activity as input or output of the activity execution: one can assign variables using the template language Freemarker.
All the variables that exist in the process can be accessed and used to assign new variables. This represents the initial variables that have been passed when triggering the process, internal variables that are automatically created during the execution and variables that have been created or reassigned in the input or output part of an activity.
D7.4 Innovative Business Process Test and Validation Report

Figure 13: A launched process input variables

Figure 14: A launched process computed variables

Figure 15: A launched process internal variables
The regular Freemarker functions library can be used in this context, as well as some specific Freemarker functions that have been developed to deal with RunMyProcess' specific environment.

2.1.3. Process and Web forms interaction: Launch Forms & Manual Tasks

One of the key features of the RunMyProcess platform is the possibility to link a web interface to a process and vice-versa. To summarize, a web interface launching screen can trigger the execution of a process and a process can generate manual tasks.

A process can be launched from a web interface if the web interface design contains a Launch Process button.
If the connected user clicks on the button, a message confirming the launching of the process will appear.

The triggered process has been launched with initial variables: all the variables from the web interface.
Manual tasks are generated by a process execution. They are usually designed as a screen of a web interface. A web interface is as described earlier, a set of widgets. Configuring a screen means configuring a partial view of a web interface by setting widgets to be visible or invisible, active or inactive.

A screen is created by default along with the web interface: the launching screen, but the web interface can have several screens.

Picking a screen from a web interface configures a manual task.
Once the process has reached the manual task activity, a manual task will be generated: the widgets displayed will match the design of the selected screen while the data displayed will be filled with the variables flowing through the process execution. The process execution will be stopped on the manual task activity until the task gets validated.

A manual task should implement at least a Continue button. This button as it appears in the manual task can be clicked to validate the manual task and resume the process execution.
A screen is really only a view from a web interface, and the web interface can be seen as a complete document that flows through the process execution.

Last but not least, all access rights to a web interface launching screen or a manual tasks are to be strictly configured.

The right to open a launching screen and to launch a process is defined at the project level whereas the right to open and validate a manual task is defined by the assigning of the manual task that is configured directly in the process.

2.1.4. Other modes to trigger a process: Scheduling and Email

A process can be triggered through a web interface as seen previously but also through other means that may ease the integration with third-party systems. A process can be triggered by sending an email to a predefined address which is a shared mailbox dedicated to the process of triggering emails.

process+{account_id}.process_id@runmyprocess.com

Once downloaded, emails are parsed to retrieve the variables that are going to be passed as input parameters to the launched process. As an example, the subject, content, cc list of recipients and attachments can be considered as input of the process.
A process can also be triggered by a predefined scheduler that is going to launch a process at a predefined date and time and eventually repeat this trigger with a defined period.

Lastly, a process can be launched directly via WS. For each process designed on the RunMyProcess platform, an API is automatically created to launch that process.
2.2. Integration in RunMyProcess

The second main feature of the RunMyProcess platform is its capability to connect to third-party exposed Web Services through the use of predefined connectors.

- Connectors can be called directly from processes.
- Calls can be parameterized using the Freemarker template language and use variables from the process.

Processes on the RunMyProcess platform can be considered as a combination between pure workflows and integration to a third-party using connectors.

In this section, we will describe all the integration mechanisms that can be leveraged in the RunMyProcess platform in order to connect the world of connected things to the current web.

2.2.1. Legacy Web Service connectors

Making the legacy Web Services available is at the core of the RunMyProcess platform. As previously stated, all objects that are made accessible using a supported protocol are then already usable in a process.
Below are the supported protocols, and data formats (media types):

- HTTP/HTTPS
- FTP/FTPS/SFTP
- SMTP/SMTPS
- XML, JSON, CSV and other character separated values, key/value, and plain text

A key issue when discussing integration with a third-party provider in today’s world is authentication. Requests made using the supported protocols, can be authenticated using one of these methods (available depending on the protocol):

- Login/password
- HTTP Basic
- AWS S3 / Fujitsu SOPOS
- Custom HTTP Authentication header
- Google 2-Legged OAuth (useful to access Google Apps API)
- OAuth 2.0

Providing several protocols and authentication methods makes the RunMyProcess an integration tool per se that can easily connect different 3rd-party providers.

To configure a connector on RunMyProcess, a designer must configure two resources: a provider that specifies the Host URL and the Authentication protocol and the connector per se that defines the service URI, the content of the calls, etc. A provider is generally shared among several connectors: it represents the configuration that can be shared among the different connectors to the same third-party.

As an illustration, let’s examine the configuration of a connector that retrieves all the Google Drive files of a user.

![Figure 27: Provider Configuration](image-url)
In this example, the authentication protocol defined in the provider is OAuth2. The access token used to authenticate the call is a parameter. To use the connector in a process, the designer will need to make sure that a valid access token is provided. This configuration makes the connector highly reusable.

![Connector Configuration](image1)

**Figure 28: Connector Configuration**

### 2.2.1.1. Connectors pre-developed library

To ease and accelerate development, the platform provides a connector publication feature. If a developer is satisfied with a connector that he/she has developed, they can decide to publish it in a global repository accessible to every account. Developers on a different account can then download the published connector and use it directly for their project.

![Public connector library](image2)

**Figure 29: Public connector library**

Connectivity can be open-sourced by using the connector publication mechanism. Currently, there are more than 2500 published connectors.
2.2.2. CoAP connector

The scope of the connectors has been extended as part of the IoT6 project to include the CoAP protocol. This has been realized by integrating the Californium java library and the version 12 of the CoAP specifications. This integration has been developed fully in the deliverable D6.5, however, it can be summarized that CoAP connectors are considered as any type of connectors and can be used in any type of process or Composite API.

![CoAP connector](image)

**Figure 30: CoAP connector**

![CoAP connector configured in a process step](image)

**Figure 31: CoAP connector configured in a process step**
2.2.3. Composite API

Composite API is an addition made in the RunMyProcess platform as a result of the exploration work done for Task T6.4. Prior to this, the only executable components that were callable were processes.

The problem with calling a process and then retrieving the response is that the caller has to poll the process status, and only when the process has ended its execution, is the response available. This is due to the asynchronous nature of a process execution. This is usually not a problem for a classical web component like a web application, that can use AJAX calls easily, but it can become a real problem for small components like the one IoT6 is targeting.

Basically a Composite API allows the user to have access to all the power of a process (Web Services integration, graphical workflow designer and all the power of a true development language at each step of the process execution), but with two additional advantages:

- The process execution is synchronous, so that the caller will receive the response within the same request that started the Composite API.
- The result of the Composite API can be transmitted like any other web API in the body of the response, using some standard formats, with standard HTTP result codes and standard HTTP headers.

In addition, Composite APIs can also be considered as a next step in the integration capabilities of the RunMyProcess platform. Composite APIs can be used to combine different connectors and expose a new API for this combination.

In the example above, two Web Services calls have been composed to offer a meaningful API: the call to the Google Login API is immediately followed by the call to retrieve all the events of a user’s calendar. This last Web Service call cannot be performed without having retrieved an authentication token from Google, but the value offered is indeed retrieving the list of events. By composing those two calls in a Composite API, we expose a new, simpler API that abstracts the authentication mechanism required from Google.
2.2.4. CoAP Proxy

We have seen that the RunMyProcess platform could integrate "outbound" with connected objects that exposed CoAP connectivity. On the other hand, we have also developed during the IoT6 project an "inbound" CoAP integration.

The solution that we have implemented is based on an independent CoAP proxy module that will proxy any call to a Composite API, making all the protocol/data format transformation needed to make the calls understandable by the process engine.

As described in the D6.5 deliverable, we have used the Californium (again, CoAP 12 draft branch), to leverage the experience acquired when developing the CoAP connectors' integration, and uses the CoAP proxy features provided by the library.

Therefore, any object that can perform CoAP calls (draft 12) can communicate with the RunMyProcess platform and launch Composite APIs, which can stand for virtually any process or combination of connectors, including CoAP connectors.
3. Web of Everything - A Platform for Composite Business

3.1. The Ever Expanding Web

Over the last 20 years the Web has had a profound effect on the way we live our daily lives and conduct business. From its roots in allowing us to share information and externalize transactions, it has continued to grow and encompass the integration of social, mobile and physical nodes. From this perspective it has often seemed as if the potential of the Web to deliver transformational new business models is unbounded. In this context, the offering and consumption of digital and social services in new kinds of IT driven value Webs should be a natural development.

At the same time, however, the technology required to participate in such transactions has remained expensive, requiring a broad range of specialist IT skills to acquire, implement, integrate and manage. This problem has only become more acute in recent times as social, mobile and physical capabilities continue to bring new and divergent technologies into the ever expanding scope of the Web.

Thankfully, however, we believe that we are seeing two major shifts that will enable organizations to more easily digitize and socialize their capabilities in order to capitalize on the transformational new business model opportunities available:

- The Web as a de-facto integration platform: Over the last few years the Web appears to have won the battle to become the de-facto platform for integration. Universal standards such as Internet Protocol (IP) networking, Hypertext Transfer Protocol (HTTP) and Representational State Transfer (REST) have provided a simple and highly scalable foundation for the consistent integration of services across organizations, geographies, technologies and node types; and

- Technology simplification via cloud platforms: Technology is gradually being integrated, simplified and made available as a higher level service, reducing technological and skill barriers and leaving organizations free to focus on their business specializations. We believe that this simplification extends the opportunities for organizations to implement new and exciting business models by providing much higher leverage tools for the delivery of services and their composition into new digital and social value chains.

Together we believe that these developments are accelerating our journey to the Web of Everything – a Web in which any kind of digital, human or physical entity can have an identity and therefore be seamlessly aggregated into new kinds of converged digital and social supply chains. We call these new value webs Composite Business Ecosystems.
3.2. Platforms for Composite Business Ecosystems

As increasing numbers (and types) of services can be easily discovered, consumed and aggregated over the Web so we believe that businesses will become more focused in scope to leverage the benefits of specialization. This hypothesis originally springs from the theory of transaction costs postulated by Ronald Coase, where falls in the costs of procuring services via market mechanisms can result in greater outsourcing. We call such networked environments Composite Business Ecosystems to represent the key concept of new enterprise structures composed from mediated ecosystems of specialized providers.

Our vision is therefore, to create a platform that provides all of the technical and business capabilities required to enable Composite Business Ecosystems implemented across the Web of Everything (Figure 33: Composite Business Platform). This platform uses the standard protocols and architectures of the Web and is intended to enable the rapid capture, execution, and monetization and sharing of business IP captured in digital form (i.e. across a combination of APIs, business processes and end user applications). In this way, we aim to foster new kinds of business service ecosystems that enable both specializations within existing industries as well as the emergence of totally new cross-industry business models.

To date, our focus has primarily been on realizing our vision within the scope of the core Web – that is the composition of informational, transactional and social services using the infrastructure of the Web.

At this stage in the development of the Web of Everything, however, we believe that the Internet of Things has the potential to be the most disruptive and dynamic element, with changes in technology, business models and society delivering new possibilities. However, based on our experiences to date within the traditional web, we believe that the full benefits of this disruption can only be realized by using the full power of both the Web and the Cloud together and in concert. For this reason, we have been focusing on understanding the ways in which we can extend the reach of our cloud platform into the Internet of Things in order to bring the full power of rapid, agile and scalable business delivery to this new area of endeavor.
3.3. Extending to the Web of Things

Over the last few years, there has been significant and growing interest in the Internet of Things. In this model, so called “smart objects” such as sensors, consumer appliances and industrial equipment can be connected to the Internet to enable them to communicate. By some estimates, the number of such connected devices is expected to rapidly grow to the order of 50 billion by 2020 and the subject is of significant interest to many national governments.

The promise of the Internet of Things is great – networks of smart devices able to constantly assess the state of some aspect of the physical world and communicate to optimize human, social or industrial activities and processes. On the other hand, the challenges of connecting the huge diversity of endpoints can be significant.

To overcome these challenges, the Internet Engineering Task Force (IETF) has been considering how to extend the reach of IPv6 for use within low power networks of constrained devices. This activity has resulted in 6LoWPAN, a new standard enabling the use of IPv6 in Low-power and Lossy networks.

While 6LoWPAN brings IP connectivity at the network level, it does not deal with the need to create an open architecture at the application layer – a prerequisite to achieving our goal of converged service ecosystems. One natural way to unify application-level communication between the virtual and physical worlds is to reuse existing Web technologies and standards. Such an approach creates a Web of Things that acts as an extension to the traditional Web. In the Web of Things, devices move beyond the IP enablement brought by 6LoWPAN and become nodes which are able to communicate and interoperate freely with all other resources on the Web.

While it has been demonstrated that existing Web architectures and protocols can successfully be used for the integration of constrained devices, such usage can already be accommodated by our existing platform and therefore, the approach is of limited interest in our activities. Additionally, however, we believe that it is unlikely that such an approach will be suitable for the whole range of limited capability devices which will eventually need to be connected.

To address these issues, the IETF Constrained RESTful Environments (CoRE) Working Group has been focused on introducing the Web interaction paradigm in a way which explicitly recognizes the differing characteristics of low capability devices. The resulting REST-based Constrained Application Protocol (CoAP) attempts to create a balance between consistency on the one hand and recognition of the limited power, reliability and bandwidth available to Wireless Sensor Networks (WSNs) on the other. Such recognition comes in the form of UDP transport, minimized message overheads and support for asynchronous interaction models. Early tests appeared to demonstrate that bandwidth, power and lifetime benefits may accrue through taking this approach and such benefits could certainly be valuable in a range of constrained environments.

As a result while CoAP is explicitly not a simple compression of HTTP, it does offer a consistent interaction model along with emerging best practices for mapping. Within the CoAP model, individual resources can continue to be identified and addressed via Universal Resource Identifiers (URIs), represented using arbitrary formats (such as JSON or XML) and manipulated using the same methods as HTTP. This use of a familiar approach has the potential to drastically reduce the complexity of developing Web-based systems that consume WSN based resources.
In this way, the proposals deliver a potentially sustainable basis for the Web of Things while simultaneously paving the way for its integration with the broader Web of Everything (Figure 34: Web of Everything).

Having studied CoAP and its utility in integrating resources from the Web of Things, we believe that it is relatively straightforward to extend cloud platform-based approaches to bridge the gap between the traditional Web and the Web of Things. More importantly, however, we believe that by becoming a mediator between the two environments, cloud platforms will solve a number of additional issues by acting as an intelligent and process based gateway (Figure 35: A Platform for Convergence) between sensors and business management systems (e.g. ticketing, and CRM, etc.).
3.4. **Benefits of a Cloud Mediated Approach to the Web of Everything**

We believe that there are a number of important advantages above and beyond speed and ease of use when integrating and orchestrating IoT resources from a cloud platform (Figure 36: Benefits of Convergence).

These benefits extend the utility of WSNs beyond simple peer-to-peer networks and enable more flexible and emergent uses of individual sensors, groups of sensors acting as peers and the wider Web of Everything.

**3.4.1. Simplification and Externalization of Function**

By externalizing complex application and process logic, individual sensors and other constrained devices can remain relatively simple and focused on their main purpose. This has the potential to vastly increase the ease of maintenance, adaptability and accessibility of applications within the Web of Everything. Rather than being forced into using complex embedded environments and strict domain models that tightly couple usage and limit flexibility amongst diverse devices, higher level applications can be created externally using mainstream tools to leverage the basic functions of constrained devices in larger contexts.

This in turn has the potential to encourage the emergence of new and unforeseen uses and therefore, support business model experimentation.
3.4.2. Composition and Abstraction

The ability to easily compose resources allows us to create “meta-sensors” - that is higher level “virtual” sensors which enable consumers to address combinations of homogeneous or heterogeneous resources - across one or more WSNs - as if they were a single entity (e.g. create a single service that addresses all lights in a building or which aggregates several different data sources to provide a more holistic view of a situation). In our view such composed meta-sensors will be a key enabler to the success of the Web of Things, as it will be critical to find a strategy for simplifying and abstracting the burgeoning number of devices without losing flexibility. Composing resources in this loose manner allows us to achieve the dual benefit of abstracting complexity for consumers where necessary, while maintaining the separation between each individual entity and the collection (i.e. to retain adaptability).

3.4.3. Resource Management

As constrained devices are brought onto the Internet, the maximum benefits of their connectivity will accrue if their services are available for others to aggregate and build upon. Without the ability to effectively manage access to constrained resources, however, there is a serious risk that performance degradation and/or power issues will result. In this context, cloud platforms can be used to create composite APIs that add e.g. throttling or caching and which leverage subscription and payment capabilities to shape consumer behavior. This will allow resource owners to manage access to their resources in a way that is appropriate to their capabilities and context in order to limit the scalability issues inherent in constrained environments. Without such easy mediation, the onus would be on resource owners and consumers to individually manage their interactions across the large web of relationships with which they are involved – a daunting and potentially impractical task.

3.4.4. Service Convergence

Supporting inbound CoAP calls from actors within the Web of Things allows us to support both intermediation between WSNs and also the presentation of any integrated SaaS or API service back into the Web of Things as if it were a sensor. Such simplification can have two benefits. Firstly, the ability to expose WSN services in a standard way removes the need for consumers to be aware of the constrained nature of the device and manage protocol switching. Secondly, it recognizes that locally deployed sensor applications that would benefit from accessing external data sources can do so without placing additional processing and power requirements into the device – effectively all cloud services look like constrained devices and can be accessed using CoAP.

We believe that this intermediation will help to reduce the barriers to integration between the virtual and physical worlds and encourage the emergence of new and unforeseen patterns of use. In addition, the combination of outbound and inbound integration will enable the more rapid satisfaction of a wide range of composition, intermediation and enhancement use cases within a consistent and high productivity environment.
3.4.5. Unified Discovery, Subscription and Monetization

As the number and variety of devices on the Web of Things continues to expand so it becomes more difficult to find, subscribe to and leverage the services provided by such devices. When viewed from the perspective of the Web of Everything, however, this problem becomes a subset of the issues experienced in finding any kind of useful Web Service to consume. Cloud platforms have been making good progress in simplifying the subscription and consumption of Web and cloud based services through the use of marketplaces. Such marketplaces can be tied into a specific platform or independent and focused on a niche such as APIs. These kinds of marketplaces could help to make Web resources easier to find – due to categorization, tagging or search capabilities – and also more consistent in terms of subscription, management and monetization. We, therefore, anticipate the extension of marketplace technology to encompass connected devices and to enable experimentation with monetization and business models for their consumption.

3.4.6. Insight and Analytics

Monitoring and managing large networks of divergent devices is set to be a daunting task, especially when they are combined with other services as part of a larger end to end “digital supply chain”. We only need to look at the complexity of the physical supply chain management to understand the difficulties inherent in such a complex undertaking – and the scope and breadth of services on the Web will dwarf this challenge. To address this issue, we need to gain visibility into the status and performance of devices and other services so that we can mitigate any impacts on the overall value chain. In this context, using a cloud platform to intermediate between services and track their performance could highlight potential issues based on response times or error frequencies. Over time the volume of data gathered about the nature of interactions within complex value webs could be used to make suggestions or undertake predictive analytics based on likelihood of failure.
3.5. **Broader Implications of Web Convergence and Composite Business Models**

We have already spoken of the way in which converged cloud platforms for the Web of Everything can be an enabler for composite business models. In this context ‘composite business’ - i.e. the ability to build businesses through the composition of value from across the Web of Everything - becomes an enabler to the rapid linkage of business and technology assets into new, consumer-centric value webs that can disrupt industry boundaries and deliver new personalized services (Figure 37: Value Chain Reconfiguration).

While these emerging shifts have the short term potential to improve services from companies and organizations operating within settled industry boundaries, the more significant business shift is the way in which high leverage cloud platforms can empower the maximum number of people through technology democratization and allow organizations to pursue wholesale specialization and the aggressive re-drawing of existing industry and social boundaries around value. We are potentially at the start of a whole new business revolution that changes the basis of competition as enterprises are able to specialize, integrate and then focus their joint efforts around value to the end consumer rather than on maximizing the utilization of their own capabilities in pursuit of scale and efficiency. As value webs can be quickly created, evolved and realigned to ‘pull’ everything into the experience required by the consumer, the old model of ‘pushing’ industrially or functionally silo-ed products and services from large and tightly integrated companies becomes insupportable.

These emerging possibilities represent both a huge opportunity and a huge threat for all businesses, industries and regions globally, since the opportunity to fundamentally change the basis of competition in an industry - and potentially to consolidate global power for a range of capabilities into a small number of Web scale organizations - could significantly affect the balance of power.
4. Innovative Business Processes

During the IoT6 project, we have implemented new business processes based on the project IoT6 use cases. Those processes have been used to demonstrate the global integration of the Internet of Things that have been made possible by normalizing the IoT6 stack.

In this section, we will describe in details the two main business processes that have been implemented and tested during the course of the IoT6 project. We will present an innovative process integrating the physical and the digital world that has been developed in 2012 for a RunMyProcess customer. Lastly, we will suggest some new use cases that could currently be developed on RunMyProcess. Those last use cases could combine the various features described in Section RunMyProcess Features and the CoAP / IPv6 integration to create innovative business processes.

4.1. Part Replacement Use Case

This use case is the first business process that has been implemented for the IoT6 project. It was partially demonstrated during the second year review.

The use case can be described as such:

- An alarm is sent by a connected object to the RunMyProcess platform

![Figure 38: Part Replacement Use Case – Alert Notification API](image)

- The alarm is logged and the alarm message is forwarded by texting a predefined phone and by emailing a predefined email address.

![Figure 39: Part Replacement Use Case – Alert Notification](image)
The person responsible for the maintenance of the building checks the connected object that has sent the alert: the device has a defect and the alert is a false positive.

The person responsible of the maintenance marks the alert as resolved on the RunMyProcess platform, which then turns off a revolving light. This revolving light was materializing the alarm in a remote office.
• The person responsible for the maintenance then scans the RFID tag of the faulty device with its phone and sends it to the RunMyProcess platform. This call is related to STIS to order a replacement part. A replacement order id is returned.

![Figure 43: Part Replacement Use Case – Creating a replacement request](image)

• The person can check the replacement order by scanning the corresponding RFID tag with its phone and forwarding it to the platform. The request is forwarded to STIS that then returns the status of the order.

![Figure 44: Part Replacement Use Case – Retrieving the order status](image)

• Once the replacement parts arrive, the person in charge of the maintenance can mark on RunMyProcess, the order replacement as completed.

![Figure 45: Part Replacement Use Case – Closing the part replacement order](image)

This use case has been implemented on the RunMyProcess prototype platform that enables the integration through CoAP with the IoT6 stack. It has been demonstrated several times during June and July 2014.

It combines the interaction between:

• A smartphone
• A third-party that exposes an API for sending text message
• A Mail Provider
• A connected revolving light
• UDG
• STIS

RunMyProcess platform acts in this scenario both as the coordinator of the whole use case and also as a tool providing the precise monitoring of the different interactions between the components.
4.2. **Smart meeting room demo (how we combine RMP features to link the real world with the web)**

The second business process that has been implemented for the IoT6 project serves as a Meeting Planning application.

The scenario can be described as such:

- **Planning a Meeting**
  - An employee plans a meeting in a RunMyProcess Web form. The employee fills in the meeting title, the meeting description and agenda, the date, start time, and end time. The employee chooses the meeting room from a pool of possible meeting rooms and the participants.

  ![Meeting Reservation Web Form](image)

  **Figure 46: SmartMeeting Use Case - Meeting Reservation Web Form**

  - The meeting is added in the employee Google Calendar. The corresponding event is entitled with the meeting title. It is located in the meeting room selected by the employee and its description is filled in with the description and agenda filled in by the employee.
D7.4 Innovative Business Process Test and Validation Report

Figure 47: SmartMeeting Use Case - Meeting Google Calendar Event

- Each participant of the meeting gets a personal RFID tag. All the tags are also associated to the meeting. They will serve as an authentication mechanism to allow the participant to join the meeting.

- Preparing the meeting room
  - 5 minutes before the beginning of the meeting, a message is sent to the smart meeting room to apply a preconfigured meeting configuration: blinders are closed, the heating is started, ...

Figure 48: SmartMeeting Use Case - Scheduled Meeting Preparation

- Accessing the Meeting
  - 5 minutes before the beginning of the meeting and during its whole duration, participants can access the room by presenting their RFID tags. The meeting room is equipped with a RFID tag scanner that requests the authorization to open the door to the RunMyProcess platform. If the scanned tag matches an authorization for the current time and concerned meeting room, the platform returns a positive authorization and the door opens. The participant attendance is logged in on the RunMyProcess platform.
• Ending the Meeting
  o At the planned end of the meeting, a process is launched to check if there is still someone in the meeting room by calling a presence sensor. If the meeting room is empty, the default settings of the meeting room are applied.

In this use case, we have used several RunMyProcess advanced features such as the internal scheduling engine, the internal MongoDB database, and processes, etc. This scenario has been implemented on the RunMyProcess prototype platform and has been demonstrated several times during June and July 2014.

To monitor the smart meeting reservation tool, we have also developed a follow-up app that is accessible only to the designer and supervisor of the tool.
RunMyProcess platform acts in this scenario both as the coordinator of the whole use case and also as the tool enabling the integration between SaaS major actors, such as Google, with connected objects.
4.3. I Heart Studios Use Case

This last business process is not directly linked to the IoT6 project. Though it does not rely on RFID tags, sensors or IPv6 for the matter, it is indeed an innovative business process that makes the link between the physical and the digital world.

I Heart Studios is a photo studio based in London. One of their main missions is to photograph clothing collections for online sellers: they receive boxes of clothes that should be photographed on models from different angles. Once the pictures are taken, they are retouched. Clothes are then returned to the customer and an invoice is generated depending on the number of shots taken.

The heavy flow of clothes in the studio has proven to be impossible to handle manually. Therefore, I Heart Studios has chosen the RunMyProcess to develop a business process that enables the clothes flow to be monitored but also to be tracked on their Salesforce account. This business process has been live since November 2012 and is a perfect example of a meaningful integration between the real world and the digital world.

Let's go into more details in the business process itself:

1. Several boxes of clothes are received from a customer. Those boxes match a corresponding shooting campaign that is represented as a job in Salesforce.
2. Each box is checked in and receives a unique barcode.
3. The boxes are opened and each piece of clothing receives a unique barcode. For each box, a box object is created on the fly in Salesforce.
4. The barcode for each piece of clothing is associated to a description previously provided by the customer. Again, each piece of clothing is created on the fly in Salesforce.

![Figure 54: Boxes Check-in Web Form](image-url)
5. The content of all the boxes is compared to the list of clothes provided by the customer: if a piece of clothing is missing, the customer gets notified.

6. Each piece of clothing is hanged on a rail that is also identified with a barcode. Each rail is assigned to a photograph and a retoucher.
7. For each rail, a piece of paper is printed that lists all the items hanging on it, the photos that should be taken, and the photograph assigned, etc.

8. A photograph gets a rail of clothes to photograph and a RunMyProcess manual task. This manual task lists all the shots that should be taken and under which name to save those shots.
9. Once the photographer has photographed all the clothes of a rail, he/she validates the manual task. A retoucher is then assigned to the rail through the physical rail being brought to him/her and a RunMyProcess manual task.

10. The rail is sent back for checkout. Through the barcode scanner, the person packing up the clothes to be sent back can assign each rail to a specific customer and campaign and check that all the clothes are received and indeed sent back.

11. The retouched photos are uploaded for the customer to retrieve them and an invoice is generated based on the number of shots and retouches done for the campaign: the invoice is generated in RunMyProcess and uploaded to Salesforce.

Figure 59: Photo Shoot Web Form
D7.4 Innovative Business Process Test and Validation Report

Figure 60: I Heart Studios Boxes Check-in / Check-out process

Figure 61: I Heart Studios Rail shooting process
4.4. Potential scenarios

The previous use case and scenarios have been implemented and have been - or will be - demonstrated during the course of the IoT6 project. In addition, we have also specified some new use cases and scenarios that we think could be easily implemented in a near future. The next section will present thoroughly these new use cases.

4.4.1. Supply shortage

The goal of this use case is to automate the monitoring of some commodity supply in order to prevent any shortage of the concerned commodity.

To such extent, we will need a connected weighting scale that can transmit the weights it supports when requested. The commodity monitored should be stored on such a scale and the RunMyProcess platform could use its internal scheduler to request regularly the weight of the remaining commodity and therefore, its quantity as well.

This scheduled process could compute an estimated date of shortage based on the retrieved level of remaining commodity project. This shortage date will be updated every time a new measure of the level of remaining commodity is made.

Once the delay to the current estimated shortage date drops under a certain threshold, the person responsible for the renewal of the commodity should be notified (e.g. by email).

We could also combine the monitoring of different kinds of commodities using several scales, in order to produce periodically a summarized report of the different commodities that should be renewed.

4.4.2. RFID Kanban (Integration with Trello and Bugzilla)

In the last years, the IT industry methodology has started to shift from waterfall methodology to agile-related approaches. New project management techniques, roles and tools have appeared among them, lean methodology, stand-up meetings, scrum masters and Kanban boards.

The agile manifesto, widely acknowledged as the founding document of this change paradigm states that: Individuals and interactions [should be more valued than] processes and tools.

Today’s industry is shifting to a more decentralized workplace. In this case, how does one manage having standalone, self-organized and constantly interacting teams with partial remote work? Kanban boards are a good example: when a team is co-located, it is highly valued to have a physical white board that displays the state of the Kanban flow, but it becomes completely useless when some work is performed remotely. This issue can be solved by IoT by synchronizing automatically a physical and a digital element.

- A physical whiteboard displays in the team office the state of the current Kanban workflow. Items or tasks are materialized with cards that carry a unique RFID tag.
A digital board has also been created on a work follow-up tool like Asana or Trello.

Every time someone moves a card from a column to another on the physical board, he/she scans the card that launches the synchronization using RunMyProcess. This synchronization can be realized by enabling the RFID scanner to launch a RunMyProcess Composite API that will be able to call Trello or Asana exposed API.

Synchronization from the digital board to the physical board depends on what API or web hooks are available. Trello provides web hooks that can be used to launch RunMyProcess. In this process, a manual task can be assigned to a team member present in the office asking him/her to move the card.

Since every card movement is associated to a process or a composite API trigger, it becomes easy to monitor the Kanban flow and to compute useful metrics such as the medium time for a task or card to be resolved.
5. Conclusions

The purpose of this deliverable was to document the activities and outcomes of Task T7.3 “Demonstrating new business processes combining the various IoT6 components”.

We have presented in thorough details the two main business process that have been implemented and tested during the course of the IoT6 project. We have also presented an actual in production business process as a justification of the interest raised by IoT business processes.

We have also suggested and detailed some new business processes. The in-depth presentation of the RunMyProcess platform serves as a demonstration of the feasibility of these recent business processes.

Lastly, we have presented a vision describing the numerous advantages in having a platform such as RunMyProcess to enable IoT. This vision is a high-level reflection but we are confident that the concrete steps taken in the IoT6 project and more specifically during the Task 7.3, will serve as an argument to support it.
References

1. Salesforce web site: http://www.salesforce.com/
2. BPMN (standard Business Process Notation and Model) website: http://www.bpmn.org/
3. GWT project page: http://www.gwtproject.org/
4. Freemarker website: http://freemarker.org/
6. Agile Manifesto: http://agilemanifesto.org/
8. Trello website: www.trello.com
11. Thomas, I.S., Gaïde, S., Fedon, L.: Making IT All Work Together
12. Sébastien Gaïde, Lou Fedon, Ian Thomas IoT D6.5 report