



Horizon 2020
European Union funding
for Research & Innovation

TECHNICAL INFORMATION TO NEW OPEN CALL PARTNERS

Technical Information

1 TABLE OF CONTENTS

2	Practical information.....	3
3	Introduction to the CloudiFacturing solution.....	3
4	The Digital Marketplace	4
5	Execution engines	6
5.1	SemWES	6
5.2	CloudBroker	9
5.3	Flowbster/CBP workflows	10
6	Tying it all together: the core CloudiFacturing platform components.....	11
6.1	Central User Management.....	12
6.2	Central Billing System.....	12
6.3	Artefact repository, execution, and data service	13
6.4	Workflow and application mediator	14
6.5	Data transfer and browsing system	15
7	Resource providers in CloudiFacturing	17
7.1	CloudSigma.....	17
7.2	IT4I.....	18
8	Reusable technologies from existing partners.....	18
8.1	Remote Post-Processing – Visualization of 3D simulation results	18
8.2	Simulation Services – GPU-accelerated solvers	19
8.3	Visual Analytics.....	21
8.4	Factory integration.....	22

2 PRACTICAL INFORMATION

This document gives a technical description of the CloudiFacturing solution, in the state that is will be available for the experiments of the second open call (and the new partners from the first open call) in the CloudiFacturing project. All experiments are required to make use of this solution, with the aim of boosting their commercial and technical outcome as well as guiding the future development of the solution. This document is thus intended to give wave 2 and wave 3 partners an introduction to the capabilities in order to make initial design decisions in collaboration with the DIHs and developers of the CloudiFacturing solution.

3 INTRODUCTION TO THE CLOUDIFACTURING SOLUTION

The CloudiFacturing solution is designed to support manufacturing SMEs and their needs for advanced cloud- or HPC-based ICT solutions, both for time-limited experiments and for the commercial operation after the end of the experiments. Striving for maximal interoperability, the CloudiFacturing project and its solution, which is being developed within, integrate manifold software and hardware platforms which have emerged from previous projects and research initiatives. Through CloudiFacturing, end users (e.g. manufacturing SMEs) will be able to seamlessly access all these platforms as if it were just a single platform.

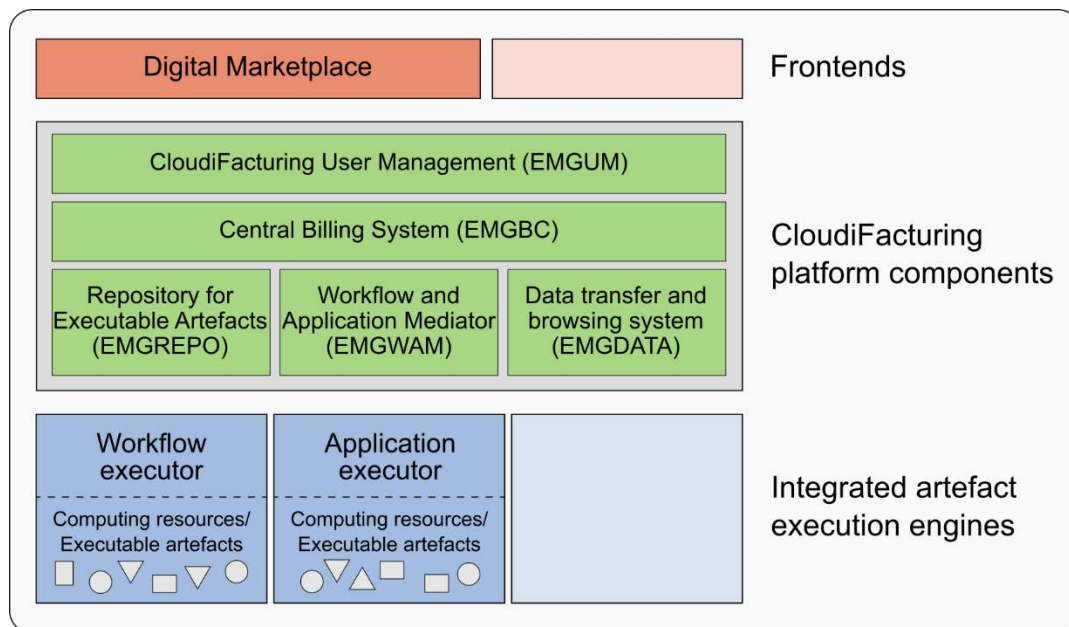


FIGURE 1: SCHEMATIC ARCHITECTURE OF THE CLOUDIFACTURING SOLUTION

Figure 1 illustrates the architecture of the CloudiFacturing solution. For end users, software deployed in CloudiFacturing will be accessible using the *Digital Marketplace* web application or vendor-specific frontends such as desktop applications or tailored web applications (red boxes). More than only a user interface to the CloudiFacturing platform components, the Digital marketplace is aiming to be a commercialization hub and space for community-building around cloud-based engineering software for the manufacturing industry. The Digital Marketplace and its offers are described in more detail in chapter 4.

The CloudiFacturing solution focuses on software packages, applications, and workflows, which are executed on cloud and/or HPC resources. Here, all these are referred to under the generic name of *executable artefacts*. CloudiFacturing integrates several different types of such artefacts and their corresponding executing engines/platforms (blue boxes in Figure 1) and will make them harmoniously accessible. Chapter 0 gives an overview of the currently supported artefact types and their capabilities. Further types may be integrated into the CloudiFacturing solution in future.

Since the described artefact types originated from several different software vendors or integrators, they are not necessarily interoperable on their own. The *CloudiFacturing platform* with its five core components (green boxes in Figure 1) homogenizes access to these artefacts and their platforms, creating a seamless "platform of platforms". *CloudiFacturing User Management* and *Central Billing* provides a single-sign-on experience across the complete CloudiFacturing solution with unified invoices for all used resources. The *Repository for executable artefacts* stores metadata of all applications, workflows, etc. deployed in CloudiFacturing, allowing for a single, *unified execution interface* used by the Digital Marketplace (or other future frontends). Since each executable artefact may use specific storage resources, a *data transfer and browsing* system is available to transfer data between all storages associated to any of the platforms. Technical details of the CloudiFacturing platform components are described in chapter 0.

Chapter 7 gives an overview of the cloud and HPC hardware resources currently available in CloudiFacturing. Finally, chapter 0 describes auxiliary services and technologies offered by existing project partners to all experiments in CloudiFacturing.

4 THE DIGITAL MARKETPLACE

The Digital Marketplace (DM) is a dynamic web application that enables customers to access the CloudiFacturing services. The DM is the reference frontend implementation to the CloudiFacturing platform and thus allows direct access and execution of all executable artefacts registered on the platform, regardless of the specific execution engine they are using.

The DM has a commercial purpose, aiming to showcase services, to directly sell services to customers, and to support the development of a community around the CloudiFacturing technology (e.g. cloud/HPC, big manufacturing data, data analytics, simulation software, engineering and manufacturing, etc.). In addition to showcasing services that can be accessed and executed from the DM, it will also be possible to explore technologies that could be provided by partners of the DM and that could be customized for the individual needs of the potential customers.

The DM provides the needed functionality to support the customers in exploiting and benefiting from the CloudiFacturing services. To achieve this, the main functionalities of the DM are clustered as (see Figure 2):

- *Information services* aim to keep the DM visitors informed, by providing general information about CloudiFacturing and its technology in form of presentations, news, webinars, etc. These services will help the DM visitors to learn from CloudiFacturing at a glance and to be up-to-date.
- *Community services* will facilitate the interaction and information exchange between community members. The community members are stakeholders related to the CloudiFacturing technology, aiming to learn more about the technology, use the technology,

offer services based on the technology, or supporters of the technology. In order to facilitate the interaction, blogs, forums, and collaboration tools in general will be available.

- *Cloud / HPC services* will support the operation of executable artefacts within the DM. This support involves the access to documentation, training, and consultancy, as well as to the configuration of software offering and computing resources. Through these services, the DM visitors will be able to operate the executable artefacts and process the generated results.
- *Financial services* aim to provide the needed functionality to manage the orders, transactions, and history of the consumed services. This set of services should create transparency and accountability for members and within multiple users of organisations.
- *User management services* will enable the organisation managers to handle the profile, the users, and the roles for their corresponding organisation within the DM. In this way, the organisation managers can define what kind of services will be available for their users or delegate this responsibility to trusted users within their organisation.
- *Security services* will guarantee the lawful and secure processing and handling of data including personally identifiable data as well as organisational data (e.g. provided data for the operation of executable artefacts or generated data through the executed services). The security services will enable the organisation to define the expected sensitivity for its associated data.

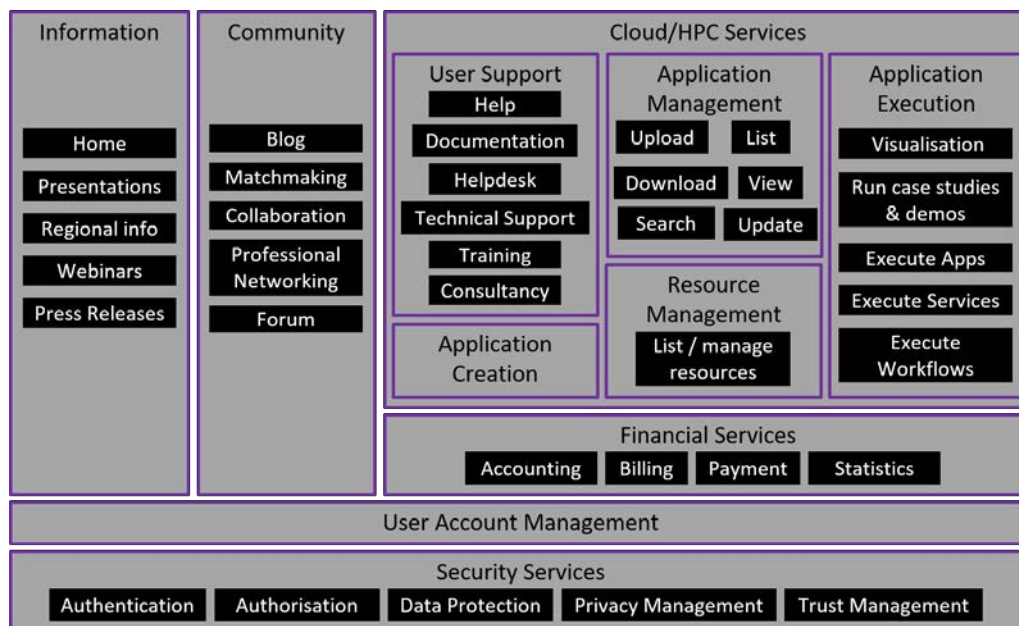


FIGURE 2: DESCRIPTION OF THE DIGITAL MARKETPLACE SERVICES

5 EXECUTION ENGINES

This chapter describes the different execution engines available in the CloudiFacturing solution. Executable artefacts developed will be deployed in one of the engines described below.

5.1 SemWES

SemWES originates from the CloudFlow EU project, the *SemWES platform* provides a workflow execution tool, where a workflow is defined as a set of calculations, simulations, or analytics tasks, which are executed for some specific user input. The scope of application of the SemWES platform are complex engineering and manufacturing workflows from simulation-based design all the way to analytics of production data and result visualization. In the CloudiFacturing project, SemWES is focused especially on computationally expensive workflows to be run on High-Performance-Computing (HPC) resources.

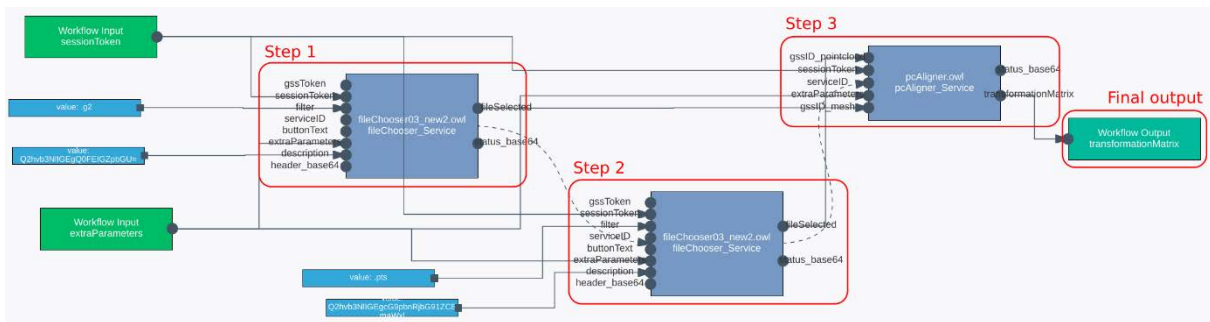


FIGURE 3: SEMWES WORKFLOW EDITOR SHOWING A THREE-STEP WORKFLOW

As shown in Figure 3, SemWES workflows are constructed via a graphical workflow editor. The shown example workflow comprises three steps, each of them representing the execution of one cloud-deployed software package called a *SemWES service*. Each service has a defined set of input and output parameters, the connections between which define the flow of data in the workflow. During workflow execution, the *SemWES Workflow Manager* automatically orchestrates this data flow and executes all steps in the right order. During the entire execution, the workflow can be monitored in the user's web browser, also when running calculations on an HPC cluster which is typically accessible only via an SSH command line. Depending on the individual execution step, the displayed feedback status ranges from simple text messages to complex and fully interactive web interfaces.

5.1.1 Selected SemWES platform services

While the flesh of most workflows will use custom SemWES services which encapsulate individual simulation or analytics software (see section 5.1.2), several generic platform services are available, serving requirements shared by most workflows.

5.1.1.1 Cloud storage access through Generic Storage Services (GSS)

With a plethora of file and object storages available in the cloud, there is also a plethora of different APIs required to access those storages. *Generic Storage Services* or *GSS* simplifies this access by providing a single, generic API which can access several different cloud storage locations. On the workflow level, SemWES-specific GSS URIs are passed as data references from service to service, while the actual data represented by these URIs is only accessed when necessary. Therefore, the flow of data

in a SemWES workflow has virtually no overhead. Furthermore, a graphical file browser using GSS is available as a SemWES platform service which can be integrated into any workflow, making upload and selection of user input a no-brainer.

5.1.1.2 Obstacle-free HPC access with the SemWES HPC service

What GSS is for file access, the *SemWES HPC service* is for accessing High-Performance Computing (HPC) resources. One of the caveats when it comes to the utilization of HPC resources is possible vendor lock-in. HPC providers usually allow login via SSH with personal user credentials, have specific queueing systems, specific locations for data storage, and possibly restricted access to cloud file storages. The SemWES platform allows the utilization of HPC resources through a generic interface, without the need to adapt to the environment and configuration of a specific HPC provider. This decoupling is achieved through the "containerization" of computation software using Singularity¹ (kind of a "Docker for HPC"), which is then executed on the HPC cluster using the SemWES HPC service. For HPC providers, a working Singularity installation on the HPC cluster is enough for integration in the CloudiFacturing platform, without the need of installing further software from experiment partners. The SemWES HPC service is fully integrated with GSS and the SemWES workflow system, making HPC access completely seamless from a user's perspective.

5.1.1.3 Flow-control utility services

A set of simple utility services is available to allow some user interaction inside workflows. For example, the user can be shown a message or asked for a binary decision simply by adding ready-made building blocks (such as visible in Figure 3) to a workflow.

5.1.1.4 Automatic input-GUI creator service

Besides input files, a workflow often requires user input in the form of several numbers, ranges, or parameter selections. To streamline such user input, a service template which generates an html input mask from a simple, table-based description of the required input parameters (including input validation) is available.

5.1.1.5 Workflow Editor and Workflow Manager

The SemWES Platform provides basic services that make it easy to integrate new services and create workflows. For the deployment of new services and the creation or modification of workflows, a graphical, browser-based Workflow Editor tool is available. The user interface of the Workflow Editor is directly integrated in the Digital Marketplace, so no installation of additional Software components is needed. The execution and monitoring of workflows is performed by another platform component, the Workflow Manager, which can also be accessed via the Digital Marketplace.

Both Workflow Editor and Workflow Manager are furthermore addressable using a SOAP API, which makes it possible to create custom user interfaces integrated with a company's own website or desktop application. All interface descriptions are also available in the standardized Web Service Description Language (WSDL), which makes automatic client generation possible.

¹ <https://singularity.lbl.gov/>

5.1.2 Custom SemWES services

The SemWES platform itself does not provide any simulation or calculation software packages, but it makes it easy for independent software vendors (ISVs) to use the platform and make their software available as custom SemWES services, which can then be integrated into SemWES workflows.

These custom services come in two flavours. Computationally demanding tasks are typically integrated as HPC jobs using the SemWES HPC service. Other tasks (including user input, pre- and post-processing of HPC jobs, data conversion, or results visualization) are commonly integrated as SemWES services which are made available as building blocks in the workflow editor. To do so, ISVs create simple SOAP web services encapsulating the desired functionality and deploy these services on the SemWES platform using Docker containers. After a simple registration procedure, the services can be used for workflow creation inside the workflow editors. Once services are integrated into the platform, they or also whole workflows, can also be offered on a commercial basis to other users within the platform.

5.1.3 Platform documentation

Independent software vendors (ISVs) wishing to integrate their software into the SemWES platform are supported by an extensive documentation platform. This platform features descriptions of all SemWES concepts and platform services as well as tutorials and code examples for all relevant areas of service implementation and workflow creation.

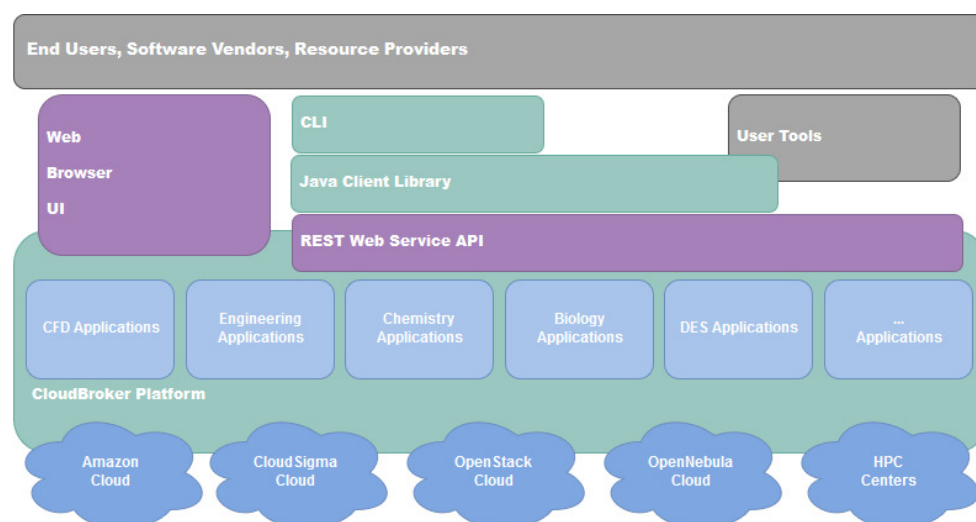
Link to the documentation: <https://github.com/CloudiFacturing/docs-and-training>

5.2 CloudBroker

One of CloudBroker's main products is the CloudBroker Platform, a backend tool for the deployment, management and running of compute intensive software on various cloud infrastructures.

The CloudBroker Platform uses IaaS from resource providers and provides PaaS to software vendors and SaaS to end users. It allows users to register, deploy, charge for and use their own cloud resources and application software or use resources and software provided by others (e.g., CB itself).

The architecture of the CloudBroker Platform is presented below.



The CloudBroker Platform incorporates adapters both to public and private cloud infrastructures, HPC centers, and both to compute and storage resources. Currently, the following cloud adapters are available on the Platform: Amazon EC2 compute and S3 storage resources, OpenStack, Eucalyptus, OpenNebula and CloudSigma compute and storage resources. Several European HPC centers can also be accessed, including the CINECA Galileo Cluster in Italy, the ROMEO Cluster in France, the ETH Euler Cluster in a Switzerland, and the HLRS NEC cluster in Germany. HPC access is possible only if the user has a separate account on a particular HPC center.

On the application side, the CloudBroker Platform is cross-domain and supports all kinds of interactive and non-interactive batch-oriented applications, both serial and parallel ones. One example of such interactive applications is ParaView⁴¹ that enables the visualization of CFD (Computational Fluid Dynamics) simulations.

The CloudBroker Platform can be accessed as a direct front-end or as a back-end middleware service. For the former, the platform is accessed with any regular web browser as a service via the internet. For frequent, advanced and automatic usage, programmatic accessibility via APIs is provided. These include a REST web service interface, a Java client library and a Linux shell command line interface (CLI). Via its different APIs, the CloudBroker Platform can be utilized by front-end software as middleware to allow access to applications in the cloud.

The CloudBroker Platform has been integrated with the WS-PGRADE/gUSE42 framework within previous projects to provide workflow capabilities. In the context of the CloudiFacturing project, its new generation, the Flowbster/CBP workflow framework with the Occopus orchestrator, will be used.

5.3 Flowbster/CBP workflows

Flowbster/CBP is a cloud-oriented workflow system on the top of the Occopus orchestrator as the new generation of the earlier referenced WSPGRADE/gUSE framework. It was designed to create efficient data pipelines by which very large data sets can efficiently be processed in clouds. A Flowbster/CBP workflow can be deployed in the target cloud as a virtual infrastructure through which the data to be processed can flow and while it flows through the workflow, it is transformed according to the workflow's business logic.

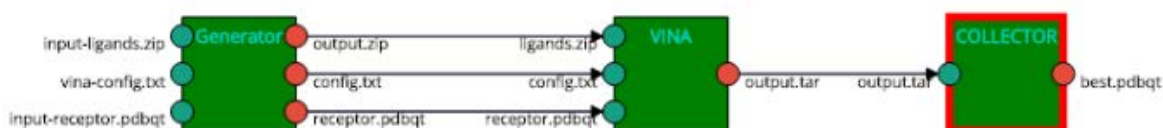
Instead of using an enactor-based workflow concept, Flowbster/CBP applies the so-called service-choreography concept where the workflow nodes directly communicate with each other. Workflow nodes recognize that they can be activated with a certain data set without the interaction of a central control service like the enactor in service-orchestration workflows. As a result, Flowbster/CBP workflows implement a more efficient data path through the workflow than service-orchestration workflows.

5.3.1 Flowbster/CBP layers

The Flowbster/CBP workflow approach is based on a layered concept. Here we distinguish four layers from bottom to top:

1. The Occopus (see next section) Cloud Deployment and Orchestration Layer serves to automatically deploy and manage the data pipeline in the target cloud, e.g. using CloudBroker Platform.
2. The Flowbster/CBP Workflow System Layer defines its uniform building block (workflow node) and execution framework by which complex data pipelines can be built. Since Flowbster/CBP is based on the service-choreography concept, the services work autonomously while communicating with other services. These services realize the functional nodes of the data pipeline workflow.
3. The Flowbster/CBP Application Description Layer enables the user to define a graph topology of the data processing workflow by defining the required number of input and output data arcs of the individual nodes and their connections, and also define the functionality of the individual nodes.
4. The Flowbster/CBP Graphical Design Layer provides an easy and intuitive graphical user interface to design the workflow layout and also to define the functions for the various nodes.

Below you can find a typical example, a 3-node workflow tested on CloudBroker Platform.



5.3.2 Occopus cloud deployment and orchestrator

Occopus is a framework that allows configuring and orchestrating distributed applications (so called virtual infrastructures) on single or multi cloud environments in a cloud-agnostic way. The configuration and initial setup of nodes can be performed through contextualization and configuration

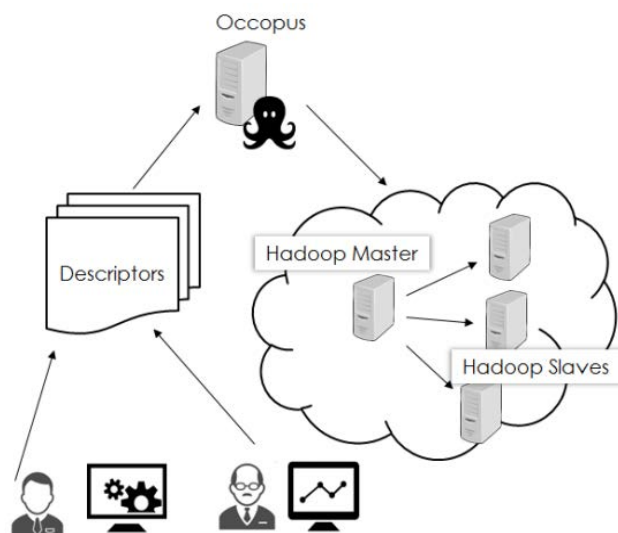
management. The built-up infrastructure can then be continuously maintained, the health of nodes can be monitored and can be recovered if needed in an automatic way.

Occopus has a pluggable architecture to handle the following cloud interfaces: EC2 (e.g. OpenNebula), Nova (e.g. OpenStack), OCCl (e.g. EGI Fedcloud), and particularly CloudBroker Platform, and CloudSigma for CloudiFacturing applications. Moreover, Occopus also has a Docker plugin to build infrastructure based on software containers. The available plugins for config managers are Chef and Puppet solo. Finally, for contextualisation cloud-init plugin is the most widespread and usable, however CloudBroker and Docker have their own contextualisation plugins as there are different contextualisation mechanisms on these resources.

Occopus performs life-cycle management of the virtual infrastructure by continuously monitoring and recovering the nodes of the infrastructure.

Manual scaling mechanism is also supported. Each node has its minimum and maximum quantity specified in the infrastructure description and Occopus takes care scaling up and down on external request within the limits. External scaling request can arrive through command-line functions or through a REST API. Automatic scaling can also be built on top of Occopus by attaching a monitoring and decision-making software like e.g. Prometheus.

Besides the above described example, another typical high level, Big Data analytical use case can be seen below, where Occopus is responsible for the orchestration of a Hadoop cluster in the cloud, and the data scientist or expert reuses the Occopus descriptors (created by e.g. a system operator) to build such virtual infrastructure.



6 TYING IT ALL TOGETHER: THE CORE CLOUDIFACTURING PLATFORM COMPONENTS

To allow for seamless access to all the different types of software artefacts listed in the previous chapter, the *CloudiFacturing platform* is being developed as a homogenization layer above these

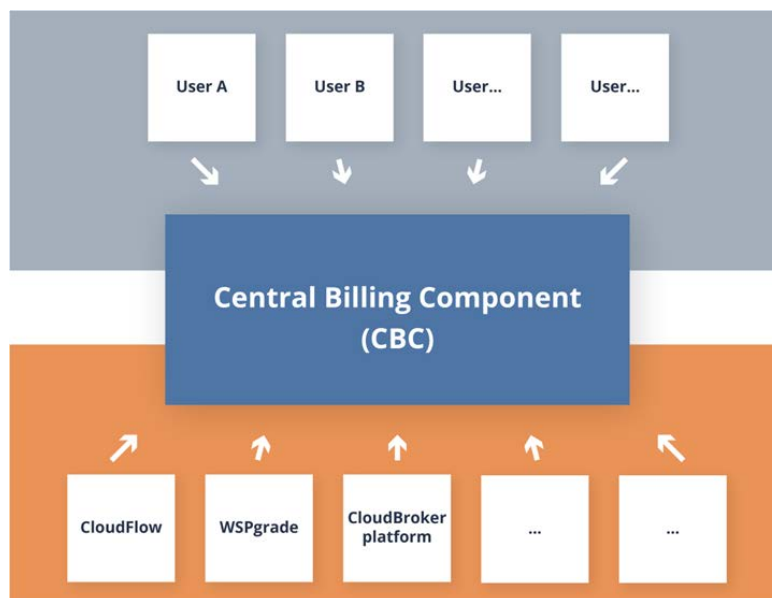
artefacts and their corresponding execution engines. The components of this "platform of platforms" are described in this chapter.

6.1 Central User Management

The main role of the **Central User Management** is to provide a common user authentication and single-sign-on solution across the platform, securing the data and computational resources. It provides a common database for all user types, including ISVs, end users and Marketplace operators. It also handles different access groups which is a prerequisite for information sharing across the platforms between different users. The Central User Management function relieves the end-user of the burden of handling multiple credentials and relieves other CloudiFacturing components from user and credentials management. In the next generation, the central user management module will also provide protection of credentials for access to external cloud resources. Experiment partners will usually not interact directly with the Central User Management, but rather rely on existing components to communicate with it and perform user authentication.

6.2 Central Billing System

The Central Billing System is provided by the Central Billing Component (EMGCBC). It will use CloudSME's AppCenter as a base technology. Billing functionality will be extracted from the AppCenter into an independent API-accessible module for the CloudiFacturing project. The general overview of the component is presented below.



The EMGCBC collects usage information from executors (SemWES Platform, CloudBroker Platform, Workflow Managers, etc.) and allows to configure different business models for the final users. It keeps track of charges based on configured business models and collected usage information. Further, the EMGCBC provides billing and usage-related reports to be charged to final users based on the business models defined.

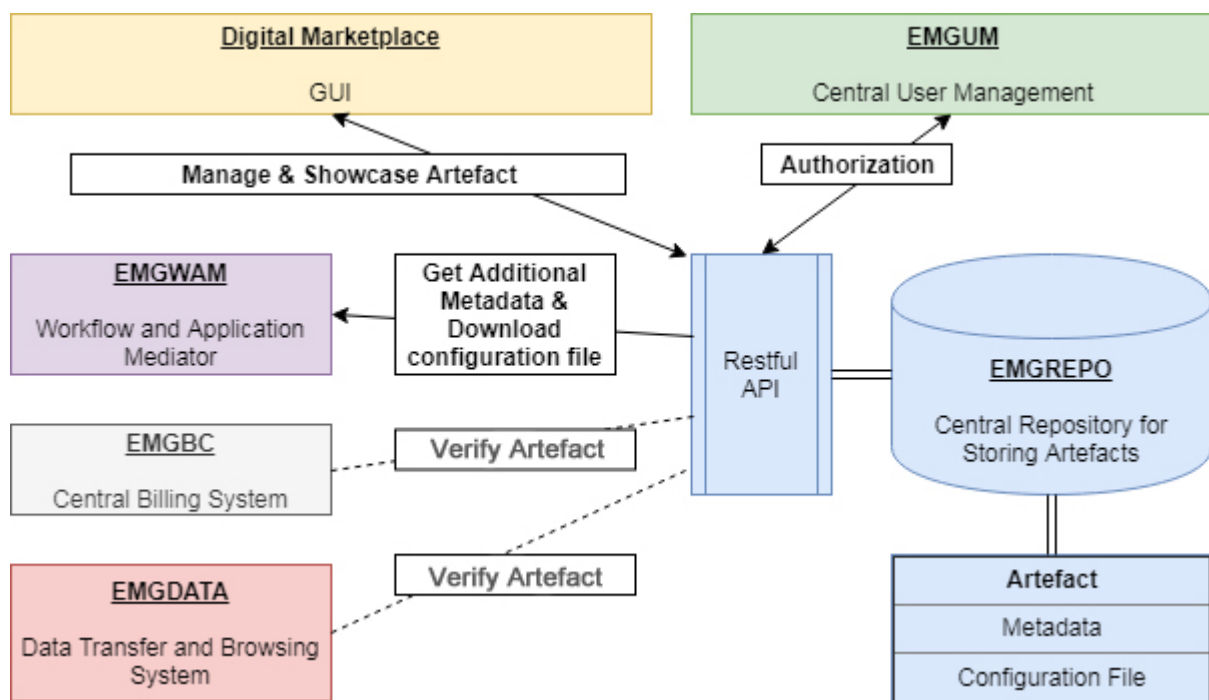
All usage and billing reports are exposed through an API which will be accessed by the Digital Marketplace and other CloudiFacturing platform components.

The EMGCBC keeps track of all the orders processed for each registered organization and all the generated bills for each organization. The history of the bills will be kept for as long as the organization exists in the system.

6.3 Artefact repository, execution, and data service

The EMGREPO component centralizes the storage and management of all the artefacts (i.e. workflows or applications) generated and used by the CloudiFacturing platform. EMGREPO stores artefacts (and service configuration file as an asset of the artefact) along with metadata in a defined directory structure. The metadata describes the artefacts and includes information such as versioning, dependencies and parameters required for running the artefact utilising the specific execution engine.

The architecture of EMGREPO and its relationship with other CloudiFacturing platform components and the digital marketplace are shown in the figure below:



As it can be seen, EMGREPO provides a Restful API for communicating with other components in the CloudiFacturing platform. This Restful API provide the following functionalities:

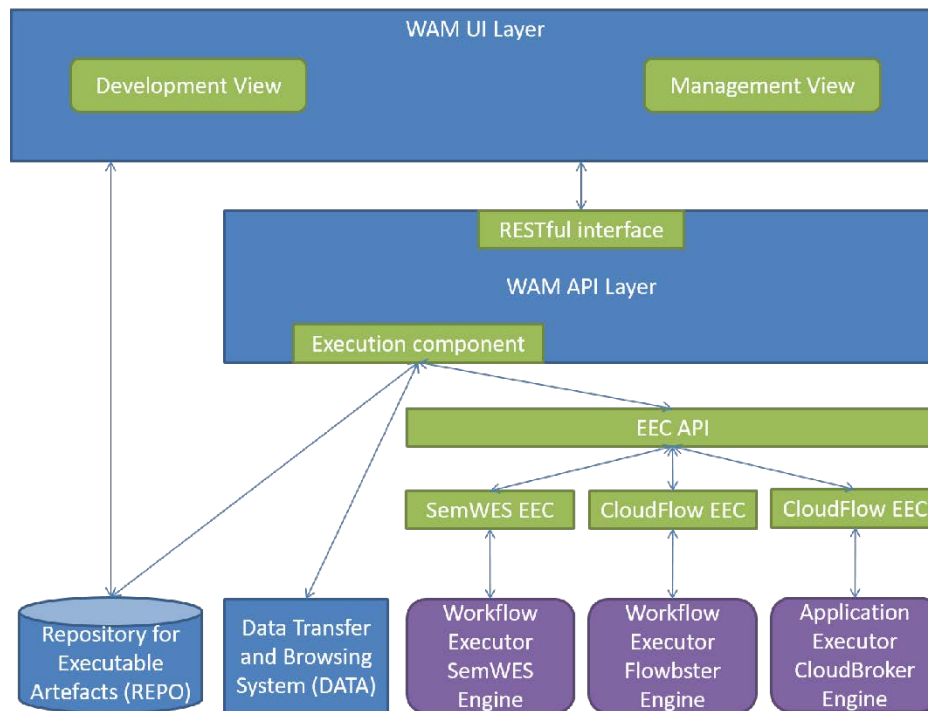
- Developer could upload/modify/search/delete artefacts through the Digital Marketplace GUI by initial API calls to the EMGREPO
- The Digital Marketplace could list/showcase services by getting the related artefact metadata stored in the EMGREPO
- End user could purchase a service from the Digital Marketplace and run the service on EMGWAM by passing through metadata querying from the EMGREPO.
- For some services, the EMGWAM could request additional metadata or download service configuration file from the EMGREPO
- The EMGREPO APIs are protected by the EMGUM

- The communications between other components (e.g. EMGBC, EMGDATA, external Execution Engine) are still under investigation

6.4 Workflow and application mediator

The task of the Workflow and Application Mediator (EMGWAM) component inside CloudiFacturing is multiple; first, it is responsible for running workflow and application artefacts from the CloudiFacturing repository, and it is acting as a meta-workflow composition tool for creating and running multi-node workflows of the artefacts defined in the CloudiFacturing repository.

The architecture of EMGWAM is shown in the figure below:



As it can be seen, EMGWAM provides a user interface for developing new workflows and managing existing ones. It also has a backend component (API Layer) for arranging the different tasks requested by the user (submit workflow, get status of workflows, cancel workflows, etc.). Access to the different workflow and application execution engines is provided with the help of the Execution Engine Clients. This architecture offers a flexible solution to extend EMGWAM with additional execution engines easily in the future.

The user interface of EMGWAM is quite task-oriented in its current form. Later the tight integration with the Digital Marketplace will offer more user-friendly and application-oriented user interfaces for running workflows and applications on top of the CloudiFacturing ecosystem.

Welcome to EMGWAM!

EMGWAM enables you to check out workflow artefacts from the CloudiFacturing REPO, and run them with the help of Execution Engine Clients.

[Submit artefacts](#) [Submissions](#)

Select a workflow to configure from REPO

The first step is to select the workflow from the repository. In the list below, you can find the names, version and types of the workflow items stored in the repository. Please select one of the and press the **Configure** button to configure the inputs and outputs for the selected workflow.

autodock_vina - 1.0.3 (flowbster) ▾

autodock_vina - 1.0.1 (flowbster)

initial_test - 1.0.0 (cloudbroker)

autodock_vina - 1.0.0 (flowbster)

autodock_vina - 1.0.2 (flowbster)

autodock_vina - 1.0.3 (flowbster)

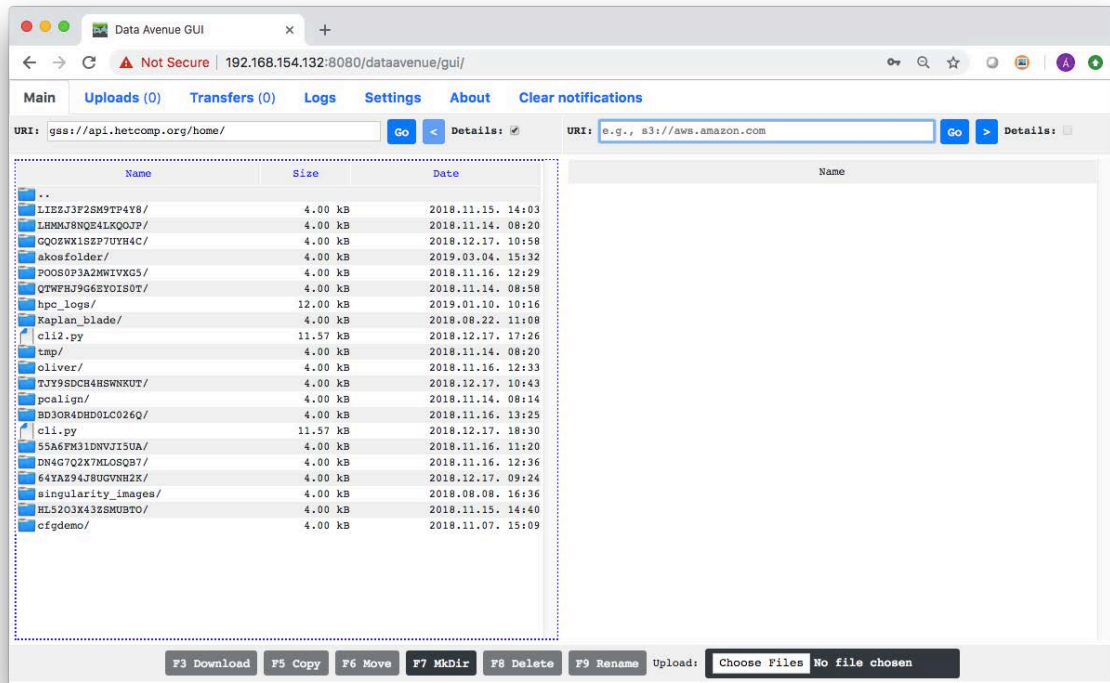
tests_basics - 1.0.0 (cloudflow)

Submissions

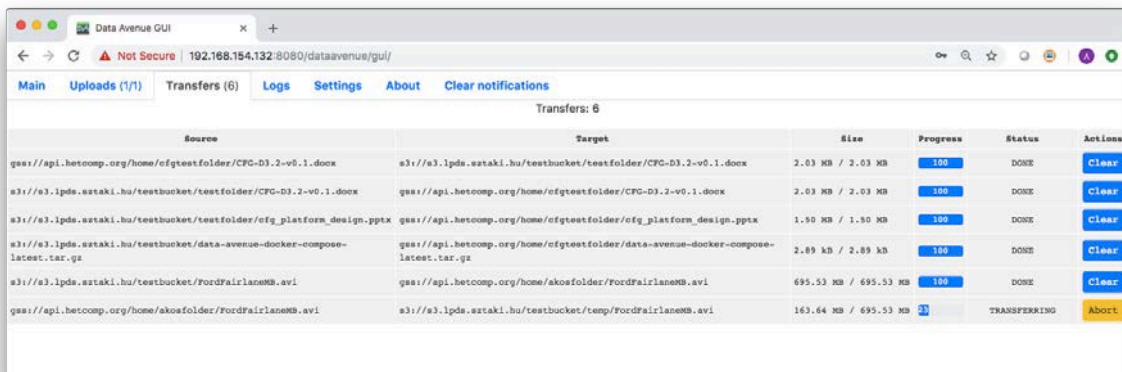
Submission identifier	Engine type	Status	Actions
f746ed8c-bb52-45c9-a97b-94b7b5c4ab6b	flowbster	1 (workflow is running on the infrastructure)	<button>Cancel</button>
bf7ef53f-1b3b-4bf4-a1a5-431dfb24a470	flowbster	0 (infrastructure building)	<button>Cancel</button>

6.5 Data transfer and browsing system

EMGDATA component (formerly known as DataAvenue) provides a service to connect to data storage resources of various types (such as Amazon S3, Generic Storage Service (GSS) for HPC storages, SFTP, OpenStack Swift, etc.) to perform common data management operations (create directories, upload/download files, delete, rename, etc.) using either the graphical user interface (GUI, shown below) or its REST API programmatically, respectively.



EMGDATA component ensures uniform management of all these diverse storage types (by hiding all technical details such as authentication types, communication protocols, conceptual differences (files-blobs-buckets-containers-directories)) and allows of transferring (copying or moving) data between any of these resources in a secure and fault-tolerant way.



EMGDATA component in this way is capable of providing an option for the end-users to upload input data for the workflows to be executed (from local disk or from another remote storage) and also to download the results (to local disk or transfer them to another persistent location) regardless of the particular storage resource used by the given application. Besides that, EMGDATA component can be used to exchange data automatically in piped applications (meta-workflows) to pass output of the previous application to the subsequent application as input (intermediate data).

7 RESOURCE PROVIDERS IN CLOUDIFACTURING

This chapter gives an overview of the hardware infrastructure available in CloudiFacturing, provided by the project partners CloudSigma and IT4I.

7.1 CloudSigma

CloudSigma is a pure Infrastructure-as-a-Service cloud provider that provides an environment with the same degree of flexibility and customisation as equivalent private in-house environments. All functionality is available via an API or WebApp. Virtual machines can be provisioned in a matter of seconds (15 sec avg.) with a high degree of control. Users have the option of full API access with all account actions available, i.e. 100% API coverage, allowing complete automation and remote infrastructure monitoring, or the option of a feature-rich, yet intuitive web browser-based GUI. The CloudSigma WebApp allows for resource management via any web browser. Project partners can select CPU and RAM to the nearest MHz and MB. VM provisioning can be achieved via a simplified wizard or a custom server creation tool.

CloudSigma provides full availability of its production cloud to all beneficiaries. Any one of CloudSigma's European locations (e.g. Zurich, Frankfurt, Warsaw) can be used. Other non-European locations can be used on request. A discounted 3-year subscription for 100 GHz CPU, 100 GB RAM, 1500 GB SSD, 3000 GB HDD has been allocated to the project. GPUs have been purchased by CloudSigma and are currently in the process of integration. These resources are dedicated to the project at no cost to project partners.

Processing, storage, networking and other fundamental computing resources are not bundled, meaning CPU, RAM, storage and bandwidth can be combined independently. This level of flexibility is available to the CloudiFacturing project partners. However, usage will be monitored to ensure the total resource limits outlined in the Grant Agreement are not exceeded by a significant margin. We will offer solutions if they do.

All cloud servers and drives are persistent and modelled on the same methodology as physical dedicated server equivalents (i.e. drives, NICs etc.). VLANs and IP addresses are also controlled using standard behaviour and support all types of traffic including multicast and broadcast traffic, which is critical for high availability infrastructure in failover.

CloudSigma's IaaS cloud platform exposes several optimisation tools allowing for greater performance levels. CloudSigma's project team will be available to assist experiment owners with configuration. For performance sensitive workloads this is especially important.

7.1.1 Software

Any x86-based operating system and software can be used with complete administration/root control including all variants of BSD, Linux, and Windows. Raw ISO images can also be uploaded and attached to CPU and RAM allowing full backwards compatibility from the platform. The end user can choose from a wide selection of ready system images including a number of BSD, Linux, and Windows-based operating systems as well as being able to quickly and easily upload their own ready ISO image. End users can customise these marketplace images using cloud initialisation frameworks such as cloudinit (see <https://help.ubuntu.com/community/CloudInit>), allowing them to take a standard installation and contextualise/customise it to their specific requirements on-the-fly first boot-up.

7.2 IT4I

There are two systems available at the IT4Innovations national supercomputing center, Salomon and Anselm.

The Salomon cluster consists of 1008 computational nodes of which 576 are regular compute nodes and 432 accelerated nodes. Each node is a powerful x86-64 computer, equipped with 24 cores (two twelve-core Intel Xeon processors) and 128 GB RAM. The nodes are interlinked by high speed InfiniBand and Ethernet networks. All nodes share 0.5 PB /home NFS disk storage to store the user files. Users may use a DDN Lustre shared storage with a capacity of 1.69 PB which is available for scratch project data. User access to the Salomon cluster is provided by four login nodes.

There are two type of compute nodes. There are 576 compute nodes without accelerator equipped with two Intel Xeon E5-2680v3, 2.5 GHz (24 cores per node) and 128 GB of memory. Salomon consists of 432 MIC accelerated nodes equipped with two Intel Xeon E5-2680v3, 2.5 GHz (24 cores per node) and 128 GB of memory. Those nodes are accelerated by two Intel Xeon Phi 7120P, 61 cores, with 16 GB of RAM.

For remote visualization, two nodes with NICE DCV software are available. Each such node is equipped with two Intel Xeon E5-2695v3, 2.3 GHz (28 cores) and 512 GB of memory. Those nodes also contain NVIDIA QUADRO K5000 GPUs with 4 GB of memory.

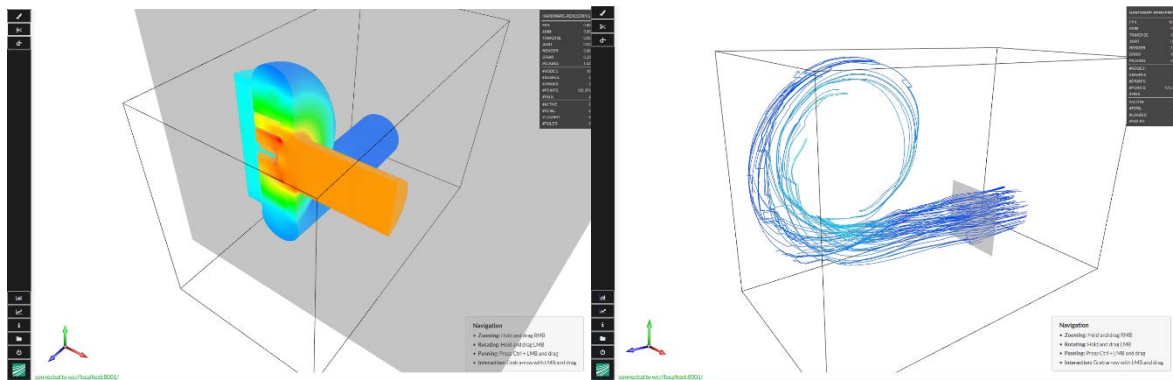
For large-memory computations, a special SMP/NUMA SGI UV 2000 server is available. This server is equipped with 14 Intel Xeon E5-4627v2, 3.3 GHz, 8 cores (112 cores in total) and 3328 GB DDR3@1866 MHz memory. It is also equipped with two 400GB local SSD disks and NVIDIA GM200 (GeForce GTX TITAN X) graphics card with 12 GB of memory.

The Anselm cluster consists of 209 computational nodes of which 180 are regular compute nodes, 23 GPU Kepler K20 accelerated nodes, 4 MIC Xeon Phi 5110P accelerated nodes and 2 fat nodes. Each node is a powerful x86-64 computer, equipped with 16 cores (two eight-core Intel Sandy Bridge processors), at least 64 GB RAM, and local hard drive. The user access to the Anselm cluster is provided by two login nodes. The nodes are interlinked by high speed InfiniBand and Ethernet networks. All nodes share 320 TB /home disk storage to store the user files. The 146 TB shared /scratch storage is available for the scratch data.

8 REUSABLE TECHNOLOGIES FROM EXISTING PARTNERS

This chapter describes auxiliary services and technologies offered by existing partners to all experiments in CloudiFacturing.

8.1 Remote Post-Processing – Visualization of 3D simulation results

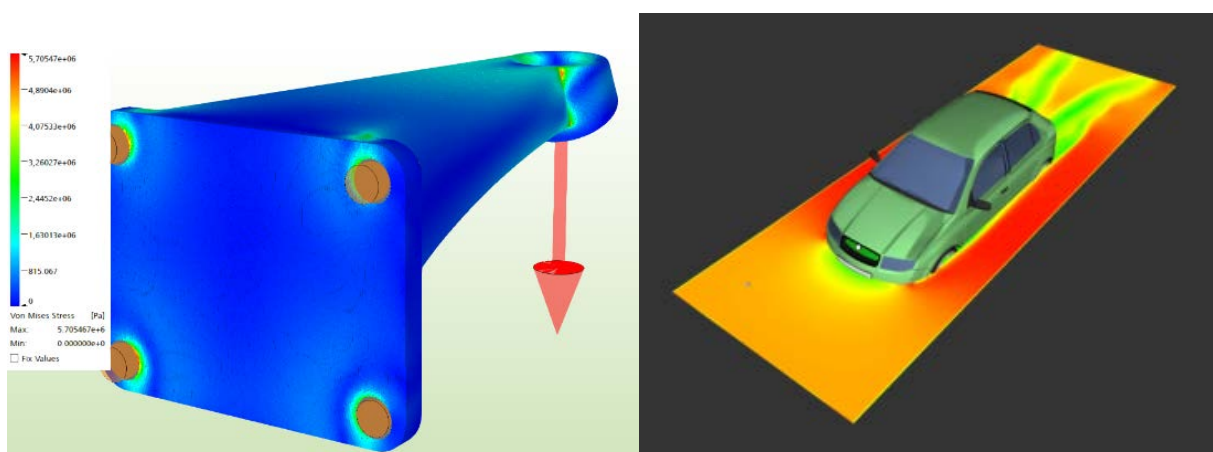


The overall goal of the Remote Post-Processing service (RPP) is to enable project partners to integrate visualization into their production process in an easy to use, fast, reliable and secure way, and therefore improve their production process.

The technology is realised via a client-server architecture in order to conduct computations on the server without the necessity to download the simulation results. The server-side computations comprise classical post-processing technology like, e.g. cross section, streamline generations or false coloring, and data transformation such that the results can be efficiently transferred to and visualised on the client. By only transmitting the necessary parts of the rendered image with some additional meta information, an efficient transmission is realised and the original simulation results can reside on the server, which might be advantageous if the original data should not be transmitted due to privacy reasons. Additionally, these techniques are combined with local (client-side) operations such as color mapping or streamline evaluation to ensure an interactive behavior on the client-side. Furthermore, the visualization techniques are optimized to support a low-bandwidth usage.

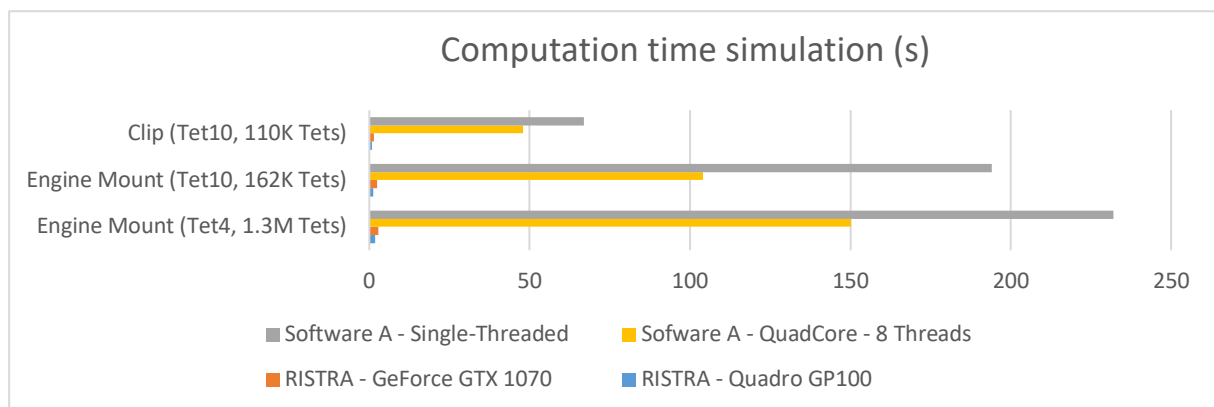
Currently, a set of visualization techniques is supported: color mapping, streamline computation, cross-section visualization, isosurface computation and value-probing. Moreover, more advanced visualization techniques will be implemented in the future. Currently, the following set of file formats is supported: VTK, OpenFOAM, and CGNS, however, with only a subset of the specifications (e.g. specific mesh types). Nevertheless, this subset can be extended, or other additional file formats can be supported in the course of the project.

8.2 Simulation Services – GPU-accelerated solvers



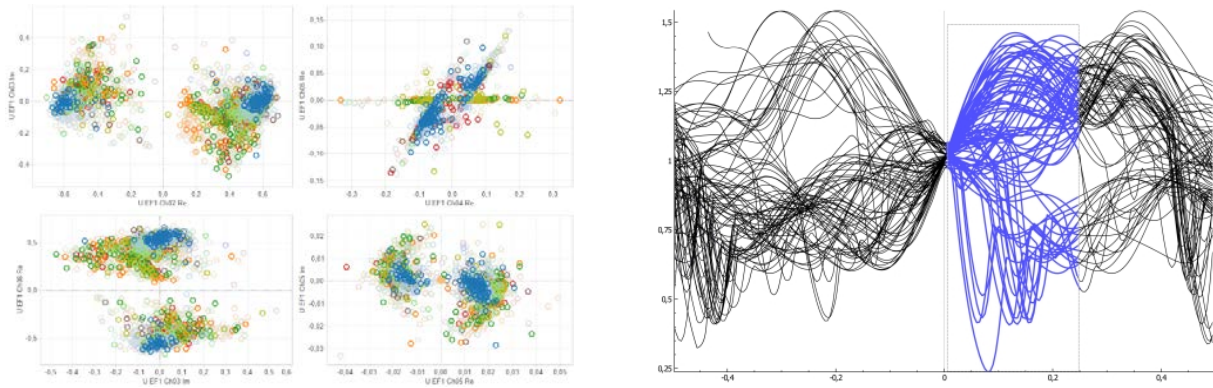
Simulation of three-dimensional physical phenomena using conventional solvers is a time-intensive task. One of Fraunhofer IGD's goals within CloudiFacturing is to develop and provide real-time support for simulation tasks using remote, GPU-accelerated solvers. Here, "real-time support" is defined as the ability to provide near-instantaneous feedback, ideally faster than real time. Real-time support for simulation enables project partners to reduce iteration times, integrate simulation into earlier steps of the product development process as geometric changes can be evaluated immediately, and use simulation to add virtual sensors to machines used in manufacturing.

Towards achieving this goal, we have developed highly optimized, GPU-accelerated solvers for structural analysis on tetrahedral meshes and (approximate) computational fluid dynamics on regular grids with cut cells. In the future, we intend to extend this further to heat transfer and modal analysis. The structural simulation solver has been validated and benchmarked extensively. It achieves speedups between 60× and 85× compared to commercially available solvers, while maintaining result quality with deviation of maximum stress and displacement of less than 1%:



By providing the solvers as a cloud service (for example as a SemWES service) with an efficient, binary WebSocket protocol, the high performance of GPU-based solvers will be exploitable without requiring local GPU resources.

8.3 Visual Analytics



Designing efficient manufacturing processes and high-quality products is a challenging task. Designs need to fulfill high demands concerning robustness, power, costs, scrap rates, durability and many others. Optimizing product and process designs can be significantly supported by the proposed Visual Analytics service. Its techniques are specifically designed to enable analysts to explore the effects of varying design parameters and to trade-off multiple quality criteria.

No matter if physical or virtual design, the design space is infinite. To improve the solutions, design options need to be carefully selected. A sound strategy for the optimization translates into the following advantages:

- Learn from *every* earlier simulation and physical prototype (failures and improvements)
- Minimize the number of simulations that need to be run and evaluated
- Minimize the number of physical prototypes that need to be built
- Sensitivity analysis to estimate tolerance levels in design

The analytical services from CloudiFacturing use the data from existing designs and prototypes to support the planning for the next improvement. This data may include:

- Design and environment parameters for documenting and distinguishing any earlier attempt for improvement
- Quality criteria derived from the aggregated simulation results (like torsional stiffness or elasticity of a construction, or throughput rate of a manufacturing process)
- Sensor data and/or quality criteria derived from data measured from physical prototypes or processes in operation

Data can be processed in the form of tables, representing a protocol of the entire optimizations process.

Automated methods like regression or interpolation models can be used to yield estimations of promising designs and promising changes to existing designs. Visualization techniques provide an overview over the different input parameters of the design space, support an exploration of how the different parameters influence the output and under which conditions, and trade-off different quality criteria with each other. Visualizations are generally based on visualization techniques for multidimensional data that are tailored in a way to cater for the specific requirements of the optimization process.

The result of the visual analysis is an overview about promising options or at least in indication of expected success rates for the implementation of a (yet) unknown design.

8.4 Factory integration

8.4.1 Introduction

Due to the growing complexity of present products and the fast variation of volumes in time and quantity, in the shop-floor the manufacturing processes are constantly affected by low-predictable events, which generate numerous interruptions and sudden adjustments of the planned production plans: normally the software used for factory integration and production planning are driven from top-level business processes and order entry, and typically they are not designed to interact in real-time with unexpected events, using in a predictive way the huge amount of machine data generated in real-time in the work-floor.

In order to support manufacturing processes with Cloud-based Simulations in a 'real-time' mode, it is necessary to:

- Develop and implement simulations of manufacturing processes.
- Use the results for setting the optimal working parameters.
- Use a software tool, which has the possibility to interface the company production-management system, to monitor the working conditions of real-time processes and machines, and to use the results of the simulation of the experiments in order to monitor the process if it deviates from optimal conditions.

In order to integrate the software tool with the real-time processes of SMEs, it is important to focus on the present architectures of manufacturing management control systems existing inside companies. The system, which manages and controls the real-time manufacturing processes is typically the Manufacturing Execution System (MES), which is a software layer between the management and planning system of the company and the manufacturing control system of machines and shop-floor (see Figure a, ISA-OSI 95 standard). The main objective of MES is to monitor the on-line execution of the production by receiving and transmitting data directly from the on-board control system of the machines, and it controls the regular execution of the orders received from the management and planning system. Figure 5 shows the general architecture of the software system,

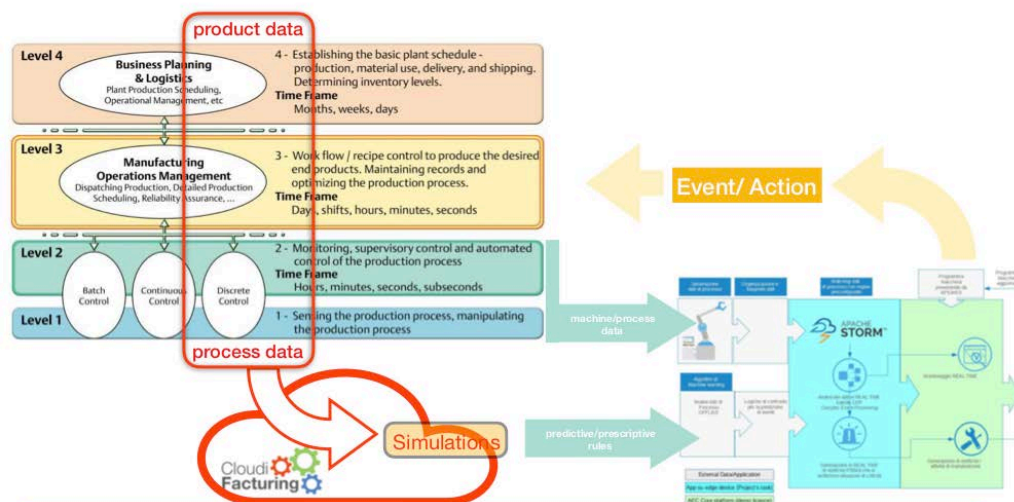


FIGURE 4: - ISA 95 ARCHITECTURE AND PREDICTIVE MODULE

which manages and controls the activities in a manufacturing company, according to the ISA 95 standard: Level 1 is the layer of the shop-floor, where machines and operators work directly on

material flow (which is usually defined as Level 0). All the data generated from and directed to machines in Level 1 are interchanged, *in real-time*, with MES (also called MOM, Manufacturing Operation Management), the software layer placed at Level 2.

MES schedules the activities and jobs to be implemented on Level 1 according to the planned customer orders to be manufactured and shipped (received from a higher software layer placed at Level 3) and confronting the proposed schedules with the real-time availability of the machines and resources (received from the lower software layer at Level 1).

Working as a middleware between Level 1 and Level 3, MES has the possibility to control and adjust, in real-time, the planned execution of customer orders, typically managed from a software called ERP, Enterprise Resource Planning, which lies on Level 3 together with other software, which controls the overall activities of a company in terms of business planning and logistics.

In terms of software architecture, the peculiarity of MES is to be the node for receiving and transmitting data in real-time from and to machines and plans.

In terms of production planning, the MES system generates and adjusts a detailed production plan, scheduling and attributing jobs to company resources, according to their 'real-time' availability.

This specific software tool, called Event-based Manufacturing Execution System (Eb-MES), couples the existing deterministic control system in use with a novel adaptive tool based on the analysis of the available field data generated from on-line processes. According to the recommendation of the project call, and the requirement received from Wave 1 experiments, especially experiment 3, this module creates control rules based on the simulation outputs and compare the optimized range set for the parameters with the on-line data stream arriving from the machines. When Eb-MES verifies that a process is deviating from the optimal range, it generates an event which consequently is transformed into a specific action, like a real-time notification to the production operators or a change in the production program, aimed to stabilize the process. Eb-MES is an additional module running in integration with the traditional control systems, and it is aimed to add an 'event-based' control to the existing deterministic one. In this perspective, end-users integrate the existing production management system, which is usually located between Level 2 and Level 3 of the traditional ISA 95 architecture (Figure 4) with the EbMES module, activating a 'bottom-up' control process generated from field data in parallel with the traditional 'top-down' system, which is activated from the operational sequence of the 'order entry'.

8.4.2 Implemented Architecture

The core of EbMES tool is based on the following components (Figure 5):

- the module which receives the real-time data stream from machines and IoT devices (Data Management and Transport)
- the module which manage a library of formalized rules (Formalized rules for Event Prediction)
- the module which compares the output of the two previous modules (real-time data stream with production rules) and generates events in case of variance (Matching of process data with formalized rules)

- the interface with the existing production management system at end-user site (Machine program from APS/MES)
- a graphic environment for the visualization of statistics, data flow, generation of events, etc.

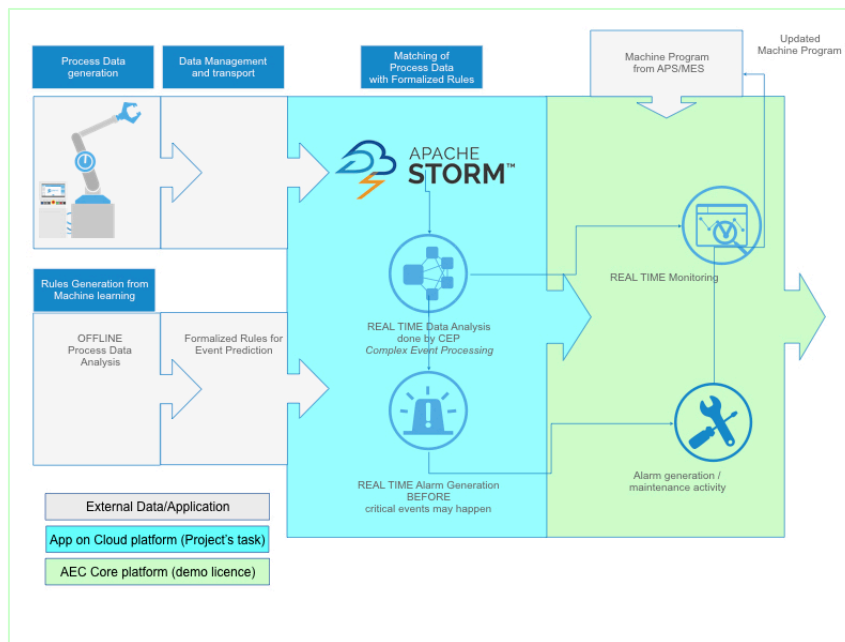


FIGURE 5 – SOFTWARE PLATFORM FOR DATA-STREAM ANALYSIS

The real-time data-stream is managed by a Mosquitto server, which collects data from different sources (PLC or sensors) and manages them with internal libraries using an MQTT protocol.

The formalized rules are produced off-line and are stored into a repository.

The core of the EbMES is the Complex-Event-Processing (CEP), which is based on a plug-in developed on the Apache Storm server. The main feature of CEP is that it can manage time-based rules, avoiding in this way the activation of actions based on spurious signals from manufacturing line. For instance, in order to get the data from the 'Machine 1', the temperature of a specific component is monitored in a range of 90s, and its median is calculated. If it exceeds 300 °C, then a corrective action will be generated. Here is an example of code:

When one of the rules is verified, CEP generates an 'event' and an 'action' will be generated according to a list of action stored in a repository.

The action will be selected (in the second period of the project) on the base of formalized rules resident in the module (green section of Figure b) which interfaces the existing MES system at customer site.

As written, in this phase rules are generated off-line, and can be defined also from the results of simulations of manufacturing processes, and in the second period it will be implemented a complex system of rules aggregation which can be used for predictive events.

The graphic environment allows the end-user to visualize important statistics, like the number of out-of-range events (Figure 7).

8.4.3 Data Exchange

In order to implement the integration of the simulations into the company system, the data flow between the different modules is organized as follows (Figure 6):

- A. The selected machines send the monitored parameters to the Application Server.
- B. The Application Server recalls the Machine Program from the customer Production Planning System (for example from the local MES or similar software).
- C. The real-time data coming from the machines are filtered: the ones, which lie inside a planned interval are moved to an external repository, like a Control Room.

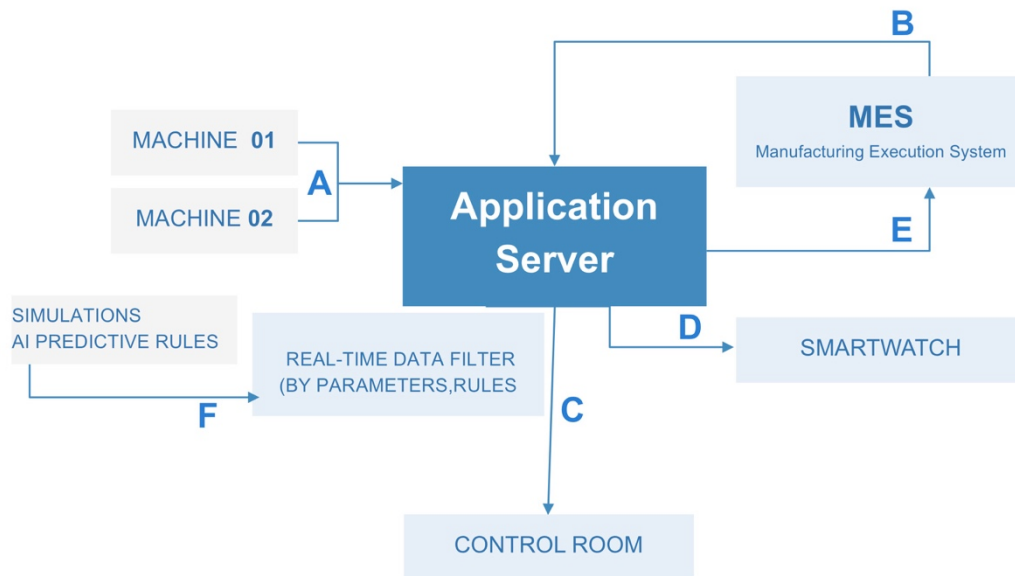


FIGURE 6 – DATA FLOW

- D. Real-time data, which don't lie in the planned interval generate an event. The event has to be matched with the relevant rule present in the CEP, and the rule generates an action (like an instant notification in a smart device or a change in the production plan).
- E. In case the rule stored in the CEP involves the prediction of a maintenance activity, the Application Server will send a message to the manufacturing planning system of the customer and it will simulate a new production plan considering a maintenance activity for the machine reserved in the negotiated time slot. During the first period of the project the Application Server runs on a public cloud, and it will be deployed on the CloudiFacturing platform on the second period.

- F. The rules for filtering the real-time data can be set up independently or on the basis of the results of the simulations.

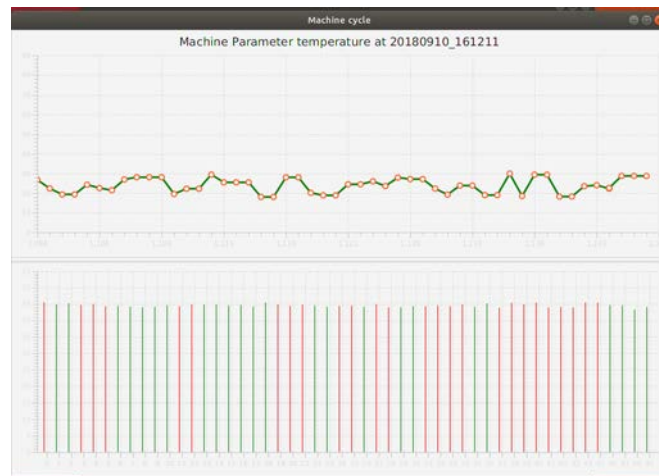


FIGURE 7 – REAL TIME DATA ANALYSIS (EX. MACHINE TEMPERATURE)

Figure 7 represents a graph showing how the monitored parameter (a temperature in this case), is continuously measured and confronted with its set value: during time intervals of 1/100 second, the acceptable values are coloured in green, the out-of-range values in red.

If the current values of the parameter don't comply with the planned rule (for example 100 out-of-range measurements within 5 minutes) then an event is created and consequently an action will be generated.

The current prototype (Figure e) collects and analyses real-time data published using MQTT protocol. The configuration used for this prototype is as following:

- Source of data is a single machine
- It's possible to define a set of rules to filter multiple process parameters
- There are one producer and one consumer, and they share the same server
- STORM applies rules and reacts triggering events when necessary
- Based on events, it's possible to generate actions for notification, monitoring, alerting or production related purposes (rescheduling, maintenance operations, etc.)

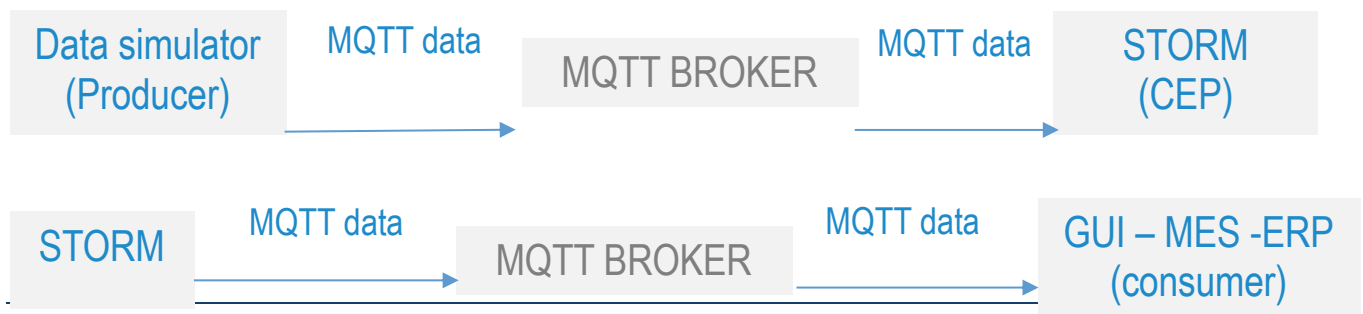


FIGURE 8 – SCHEME OF DATA FLOW IN CURRENT PROTOTYPE