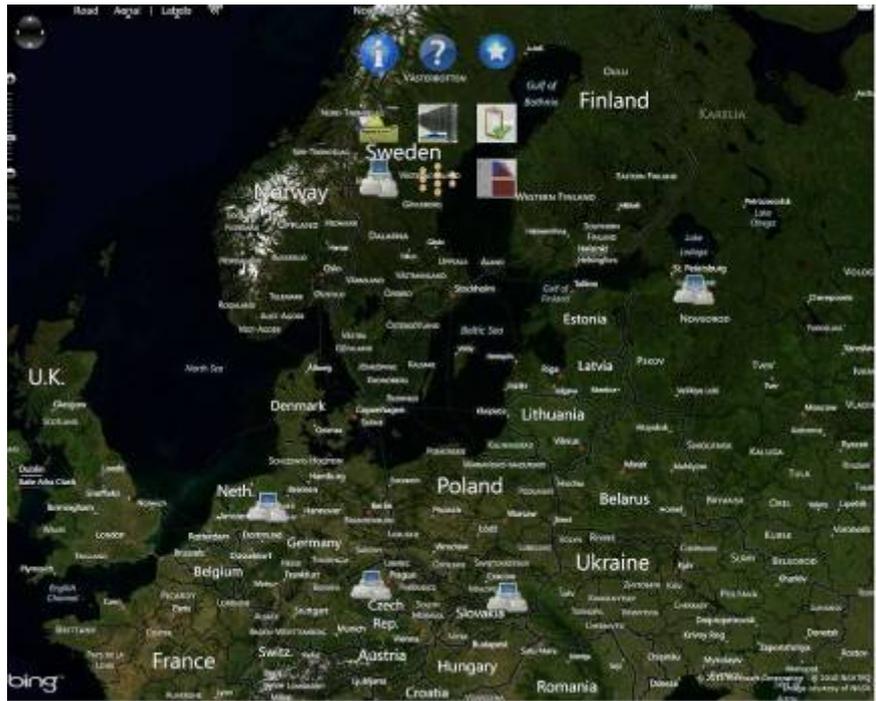


# UrbanFlood



## Running multiple EWS on distributed clouds

### Work Package 6 – D6.5

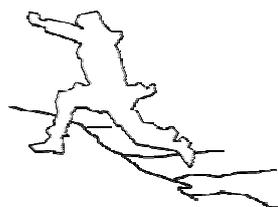
version 1.1, 29.11.2011

November 2011



## URBAN FLOOD

A project funded under the EU  
Seventh Framework Programme  
Theme ICT-2009.6.4a  
ICT for Environmental Services and  
Climate Change Adaption



Grant agreement no. 248767  
Project start: December 1, 2009  
Project finish: November 30, 2012

Coordinator

Urban Flood Project Office at TNO-ICT  
Prof dr Robert J. Meijer

Eemsgolaan 3  
PO Box 1416  
9701 BK Groningen  
The Netherlands

E : [robert.meijer@tno.nl](mailto:robert.meijer@tno.nl)  
T: +31 50-5857759  
W: [www.urbanflood.eu](http://www.urbanflood.eu)

## DOCUMENT INFORMATION

Title	Running multiple EWS on distributed clouds
Lead Author	Artem Ozhigin
Contributors	Jeroen Broekhuijsen
Distribution	Public
Document Reference	UFD6.5v1.0.SIE

## DOCUMENT HISTORY

Date	Revision	Prepared by	Organisation	Approved by	Notes
27.10.2011	1.0	Artem Ozhigin	Siemens		Initial revision
29.11.2011	1.1	Artem Ozhigin	Siemens		Updated upon J. Broekhuijsen feedback
30.11.2011	1.2	Jeroen Broekhuijsen	TNO		Added new Images to illustrate available cloud functionality. Added new text to explain the images.

## ACKNOWLEDGEMENT

The work described in this publication was supported by the European Community's Seventh Framework Programme through the grant to the budget of the Project **UrbanFlood**, Grant Agreement no. 248767.

## DISCLAIMER

This document reflects only the authors' views and not those of the European Community. This work may rely on data from sources external to the UrbanFlood project Consortium. Members of the Consortium do not accept liability for loss or damage suffered by any third party as a result of errors or inaccuracies in such data. The information in this document is provided "as is" and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and neither the European Community nor any member of the UrbanFlood Consortium is liable for any use that may be made of the information.

© **URBANFLOOD CONSORTIUM**

## TABLE OF CONTENT

<b>SUMMARY</b>	<b>ii</b>
<b>ACRONYMS AND ABBREVIATIONS</b>	<b>iii</b>
<b>1 INTRODUCTION</b>	<b>4</b>
1.1 UrbanFlood .....	4
1.2 Structure of this Document .....	4
<b>2 CLOUD COMPUTING</b>	<b>5</b>
2.1 General Description .....	5
2.2 Cloud Services in UrbanFlood .....	6
2.3 UrbanFlood Cloud Infrastructure .....	7
<b>3 RUNNING EWS IN A CLOUD</b>	<b>9</b>
3.1 EWS design .....	9
3.2 EWS deployment .....	10
3.3 Currently running EWS .....	12
<b>4 REFERENCES AND BIBLIOGRAPHY</b>	<b>15</b>

## SUMMARY

This document describes current progress of UrbanFlood distributed cloud platform development and especially features which allow for running multiple EWS simultaneously. Basic overview of cloud computing approaches are given, UrbanFlood platform design introduced. Main principles of EWS design, deployment and execution in the cloud are described and currently running EWS listed.

## Acronyms and Abbreviations

C&D	Communication and Dissemination
CYF, CY	Cyfronet AGH, Cracow, Poland – UrbanFlood partner
Defra	Department for Environment, Food and Rural Affairs, UK
DoW	Description of Work, annex to the UrbanFlood Grant Agreement
EC	European Community
EWS	Early Warning System
FLOODsite	EU funded project: Integrated Flood Risk Analysis and Management Methodologies. 6 <sup>th</sup> Framework Programme
FRM	Flood Risk Management
HRW	HR Wallingford
ICT	Information and Communication Technology
IJkdijk	IJkdijk Foundation, dike testing site in Groningen, the Netherlands
IWS	International Online Early Warning Workshop
LiveDijk	Test location for sensor technologies at Eemshaven, Groningen, the Netherlands
M&E	Monitoring and Evaluation
NEMO	Science museum in Amsterdam, the Netherlands
PU	Public (report)
RE	Restricted (report)
RTD	Research (in DoW)
SIE	Siemens, Germany. OOO Siemens in Russia is an UrbanFlood project partner
SME	Small and Medium sized Enterprise
STOWA, STO	Dutch acronym for the Foundation for Applied Water Research, Utrecht, The Netherlands – UrbanFlood Partner
TNO	TNO Dutch organisation for Applied Research, The Netherlands – UrbanFlood lead partner
UK	United Kingdom
US, USA	United States of America
UvA	University of Amsterdam
VM	Virtual Machine
WP	Work Package (there are 7 work packages in UrbanFlood)

## 1 Introduction

### 1.1 UrbanFlood

UrbanFlood is a project investigating the use of sensors within flood embankments to support an online early warning system, real time emergency management and routine asset management. It is a project under the EU 7th framework Programme which started in December 2009 and will run for 3 years. Partners of UrbanFlood include TNO Information and Communication Technology, the University of Amsterdam and STOWA (Dutch acronym for the Foundation for Applied Water Research) from the Netherlands; HR Wallingford in the UK, ACC Cyfronet AGH in Poland and OOO Siemens in Russia

### 1.2 Structure of this Document

**Chapter 2** Describes cloud computing basics

**Chapter 3** Describes principles of running multiple EWS in a distributed cloud platform

## 2 Cloud Computing

### 2.1 General Description

Cloud computing is a technology of distributed data processing allowing for shared resources, software and information to be provided to users on demand as a service. The term “cloud” is used as a metaphor based on the representation of Internet on network diagram or as an image of complex infrastructure while all technical details are hidden. Wide-spread development of cloud services was enabled by significant progress in virtualization technologies.

Cloud services can be considered on different levels of abstraction:

- Cloud Client – includes software and/or hardware which is targeted and specifically developed to the delivery of cloud services to user. It completely relies on those services which makes it essentially useful without them.
- Cloud Application Services or “Software as a Service” (SaaS) – a way to deliver software as a service by means of Internet. This way of application delivery lets to avoid any necessity to install software. Moreover it makes maintenance, support and update much easier.
- Cloud Platform Services or “Platform as a Service” (PaaS) – provide specific computer platform or solution as an Internet service. The platform can be used to host cloud applications without burden of buying and managing the underlying hardware and software layers.
- Cloud Infrastructure Services or “Infrastructure as a Service” (IaaS) – provide computational infrastructure, usually virtualization environment. Consumers rent exactly those resources which are required instead of purchasing of data-center and network equipment and software.
- Server layer – consists of hardware and/or software which are designed especially for the rendering of cloud services, including multi-core processors and cloud-specific operating systems.

Cloud can be implemented using different deployment models. It can be public (external) when all services are provided over Internet to general public via self-service interface by specific provider. In case when a number of organizations are sharing cloud infrastructure it is a community cloud. Both mentioned models can be combined to form a hybrid cloud. Private cloud can be deployed within a private network.

## 2.2 Cloud Services in UrbanFlood

The UrbanFlood hosting platform is intended to host lots of different applications which are running on different operating systems and have different demand to the computational resources. The platform is based on virtualization technologies and consists of a distributed set of virtualization servers and server pools. Thus the UrbanFlood platform provides community cloud infrastructure services (Infrastructure-as-a-Service) for running numerous VMs.

Major characteristic points of the UrbanFlood platform are that it uses online sensors, hosts multiple online EWSs, includes an online Common Information Space (CIS), uses Internet-based cloud computing services and provides online visualization, decision support and a public information framework [1]. Details of the UrbanFlood platform implementation are described in [4].

All activities within the cloud are managed by the Common Information Space (CIS, see [2]) at every stage of EWS and its components lifecycle. The whole CIS is running within the same infrastructure cloud. It is responsible for generating and launching a necessary number of components running in separate VMs and establishing data communication between them making up an EWS on demand. Thus the conjunction of the infrastructure cloud platform and the CIS creates a framework enabling the stakeholders to develop and run EWS in the cloud, e.g. providing EWS-as-a-service. Thus instead of a pure IaaS cloud platform provided by virtualization servers the target of the UrbanFlood project is a full-fledged cloud service platform specifically aimed at sensor data processing, EWS hosting and decision support tasks.

Currently the cloud is formed by resources hosted at four sites as shown in Figure 1.

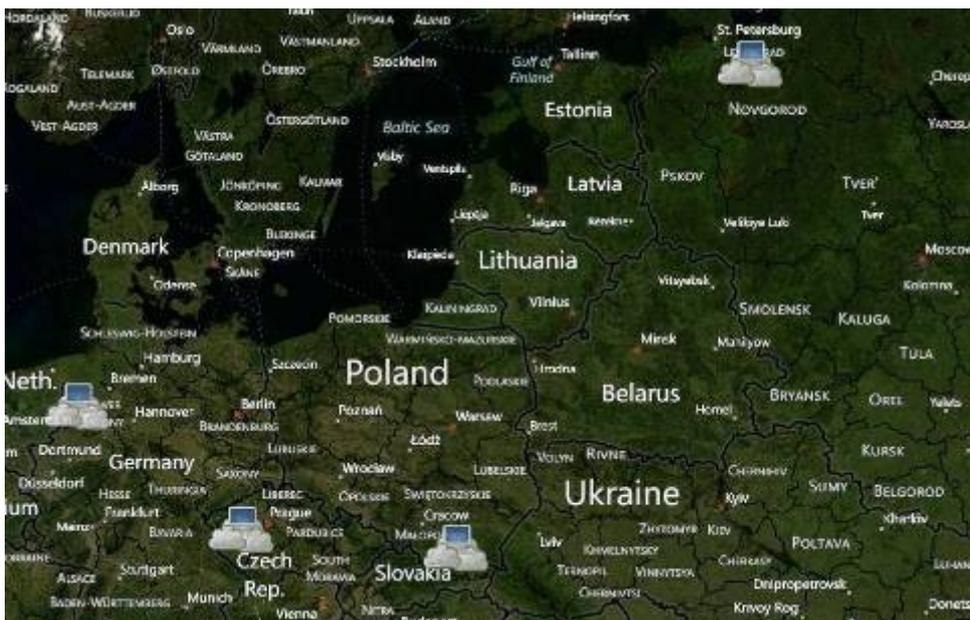


Figure 1 - UrbanFlood distributed platform sites

These sites are:

- Groningen, the Netherlands, hosted by TNO
- Germany, rented by TNO
- Cracow, Poland, hosted by Cyfronet
- Moscow, Russia, hosted by Siemens

Sites, their location and metrics can be viewed via operational Internet DSS viewer [3] at <http://cool.telecom.tno.nl/urbanfloodwebsite/index.aspx>

### 2.3 UrbanFlood Cloud Infrastructure

The UrbanFlood cloud platform is based on XenServer virtualization technology. Every component of an EWS runs as a separate virtual machine. XenServer provides extensive API for management of VMs, their resources and getting metrics. All sites are combined into a cloud and presented to the CIS via single endpoint for all operations. An outlook to the hierarchical structure of the cloud platform is shown in Figure 2.

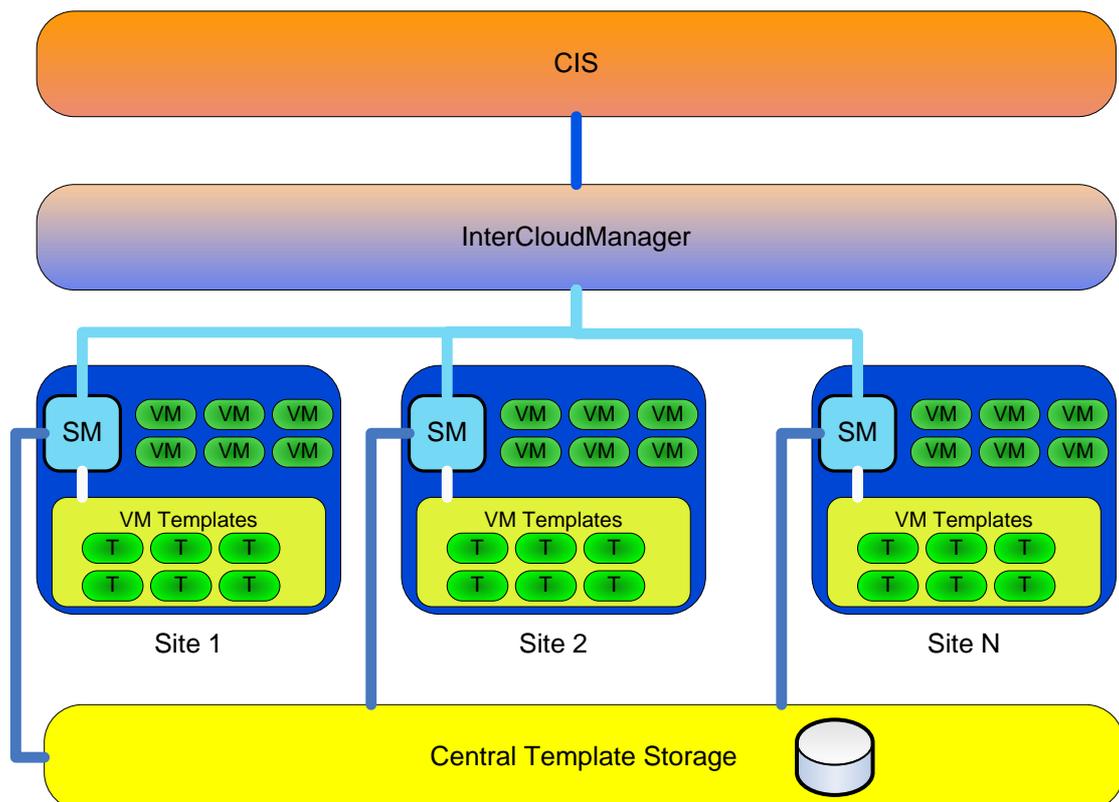


Figure 2 – Cloud platform structure

The CIS controls EWS components through a single endpoint provided by the InterCloudManager. It is responsible for dispatching necessary commands of VM management and resource allocation and collecting performance and diagnostic metrics

from all sites. Every site runs a dedicated SiteManager (SM) which carries out all VM management tasks on-site and reports metrics to the InterCloudManager. Before their deployment all components are stored in the form of VM templates in a central template storage. SiteManagers keep track of this storage and synchronize local site storages with it over a secure file transfer protocol. Every time a component is to be started a VM is spawned from a specific template in the local storage.

Although following the cloud paradigm all technical details are hidden from upper layers all the viable metrics are accessible by the CIS (as shown in Figure 3) and available for visualization, analysis and fault detection. This makes deployment of platform monitoring EWS possible within the same framework.

Encapsulation of all EWS components into a dedicated VM allows for providing all kind of applications (either legacy or newly developed one) with comfortable environment (including OS, RTE and middleware) keeping adaptation efforts to a reasonable minimum and besides that also keeps all applications separated with minimal interference to each other.

To be able to work within an UrbanFlood EWS all applications are to be wrapped into a XEN VM and then converted into a template. Such VM must contain an operating system, bootstrap code responsible for startup VM configuration based on the data provided by the SiteManager and an application wrapper allowing it to interface to the CIS and the application itself.

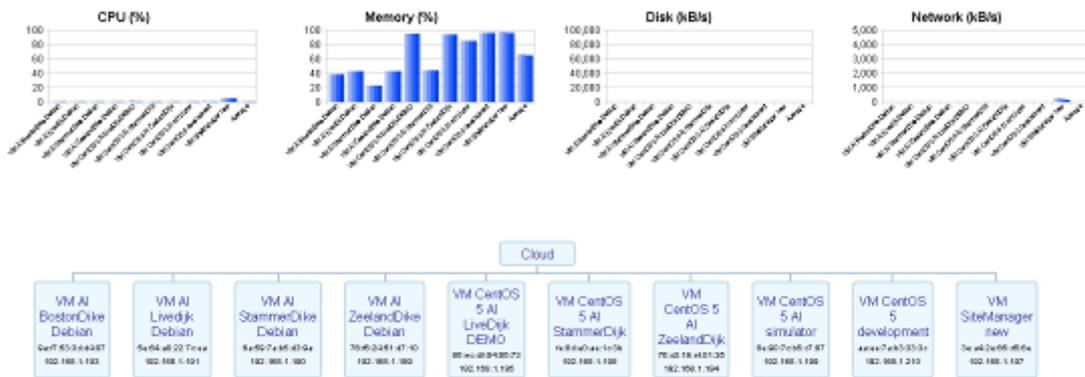


Figure 3 - Cloud site metrics

### 3 Running EWS in a cloud

#### 3.1 EWS design

Main focus of the UrbanFlood project is the development and deployment of dike monitoring early warning systems. A User shall be able to design and deploy an EWS for an object as soon as sensor data from the object are available via Internet. As the main sensor data streaming and archiving facility in UrbanFlood acts AnySense software provided by TNO. It allows for online access as to current sensor readings in real-time as well as to historical data from any specified period.

In UrbanFlood any EWS is built up out of components and EWS parts. An example of the components inside an EWS is shown in Figure 4. This figure shows the objects that can have an Early Warning System on the left, and specific functionality of the EWS in the middle. The parts needed for a specific functionality is shown in the diagram on the right. The monitoring from the SiteManagers is also shown in this graph. All of these components are up and running normally, indicated by the green box in the diagram.

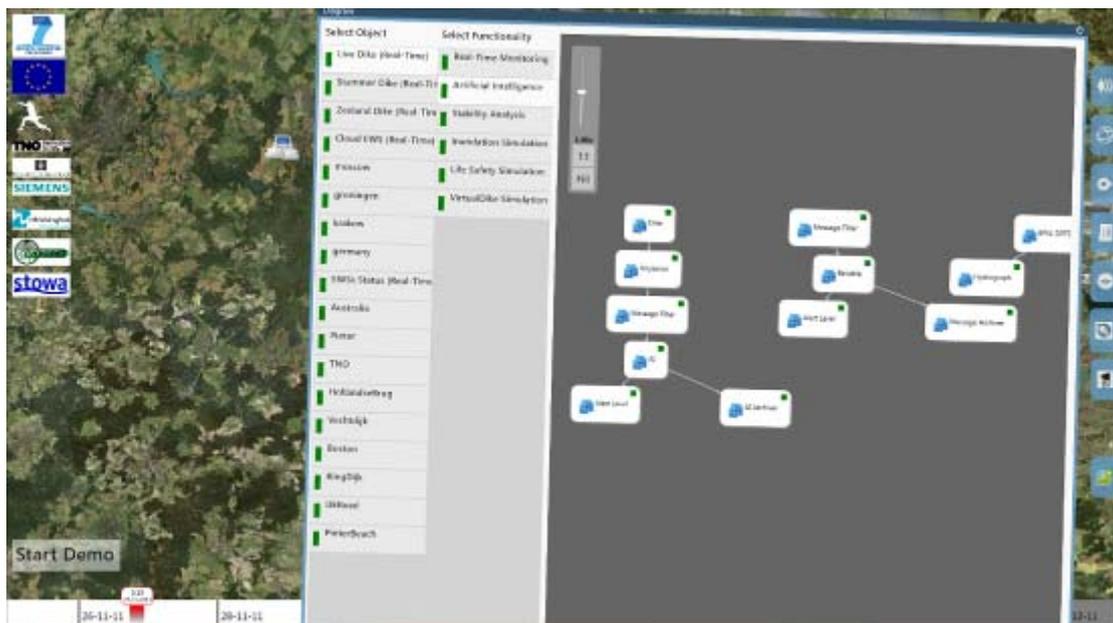


Figure 4 – EWS components

EWS parts are pre-defined, frequently used EWS fragments composed out of several components to carry out particular piece of functionality within one or several EWSs. CIS provides an interface that an expert user can use to design the structure and behaviour of an EWS and to store it as a reference model or a blueprint from which it is possible to deploy one or more EWS. The EWS design blueprint is stored by CIS together with all associated configuration data in UrbanFlood registry (UFoReg).

The non-expert user also has a view of the functionality for a specific EWS. He sees the resulting functionality on a high level basis and is able to open it and use the functionality provided by the constellation of parts. This is shown in Figure 5 and Figure 7.



Figure 5 – EWS Functionality

### 3.2 EWS deployment

The deployment process is also managed by the CIS which passes necessary commands and data directly to the cloud platform. The minimal building block of any EWS is a component (appliance) carrying out a particular function within the EWS. Every appliance (as was described above) runs in its own virtual machine. All possible types of appliances are stored in the UrbanFlood centralized storage as images of Xen virtual machine templates. All site local storages are synchronized with centralized appliance storage to speed up the deployment of any requested component. All necessary information on EWS configuration and appliance configuration is stored in a part of the CIS called UrbanFlood registry (UFoReg). Following the information received from an EWS blueprint the cloud platform deploys all necessary components. Dynamic Resource Allocation component of the CIS (DyReAlla) and the InterCloudManager based on current cloud resource metrics decide on which sites necessary appliances are to be deployed. Commands to instantiate VMs from templates are passed through the InterCloudManager to corresponding sites.

After an EWS is deployed as per its design it can be run by a command from the CIS. This command is also relayed via the InterCloudManager to SiteManagers of appropriate sites where deployed appliances are residing. All necessary configuration data from appliances are also stored in UFoReg. Along with the command to run an appliance the link to the corresponding configuration is passed to the SiteManager. The SiteManager passes this link to the appliance along with site-specific network connectivity related initialization data (IP address, gateway, etc.) which are necessary for the appliance to start communicating.

Upon appliance startup the bootstrap code within the appliance takes care of proper configuration of the network subsystem based on the initialization data and then downloads application-specific configuration file from UFoReg. This file is passed to the application which is invoked at the end of bootstrap code execution.

After an EWS is started the CIS establishes a monitoring framework for it. This framework observes cloud platform metrics which are used to correct resource allocation when the load or EWS priority is changed. Besides that the monitoring framework also keeps track on the appliance output to detect appliance failures and trigger restart for recovery.

Figure 6. shows appliance configurations stored in UFoReg. One of these configuration files is passed to a specific appliance when it is deployed as a part of the EWS to which the configuration belongs. Via these configurations any appliance can be fine-tuned towards requirements of any EWS.

The screenshot shows the 'UFoReg - UrbanFlood Registry' web interface. The page title is 'Appliance configuration registry for UrbanFlood'. It displays several sections of registered configurations:

- Registered configurations for AI appliance type:**
  - AI\_Boston (4e983f65f9180c47d3004c21) Configuration
  - AI\_Live (4a084259f9180c47d3007247) Configuration
  - AI\_Stammer (4e984273f9180c47d3007463) Configuration
  - AI\_Boston\_1 (4e9b1d8cf9180c27ed003833) Configuration
  - AI\_Stammer\_1 (4e9b2b75f9180c40420029a7) Configuration
  - SF4 (4e9c274af9180c18d6014165) Configuration
  - ASMG1 (4ea0292df9180c15bf004d9d) Configuration
- Registered configurations for AnySense Message Generator appliance type:**
  - CIS demo SDG config (4ea56b01f9180c3c73002b5e) Configuration
- Registered configurations for Flooding simulator (DRFSM) appliance type:**
  - to\_be\_removed (4d7f7825f9180c2aad017daa) Configuration
  - live-dijk-tbr (4d92f211f9180c79b0077328) Configuration
- Registered configurations for Probability Calculator appliance type:**
  - CIS demo PB config (4ea56b21f9180c3c70002c5c) Configuration

At the bottom, there is a section titled 'Upload a new appliance configuration' with a text input field for 'Configuration name'.

Figure 6 - Appliance configurations in UFoReg

At the moment there are several types of components developed as AI by Siemens, Reliable, Hydrograph and DRFSM by HR Wallingford etc. There are no limitations implied by the cloud platform itself on the number of running EWS as soon as resource constraints are not exceeded. So, running multiple EWS on distributed cloud platform is reality now. Although not all components are supporting dynamic reconfiguration and can be deployed in a fully automated way several EWS are already running bringing practical results. Figure 7 shows the multi-touch table visualization of running AI component results from two EWS-es monitoring dikes in Eemshaven (the Netherlands) and Boston (UK).

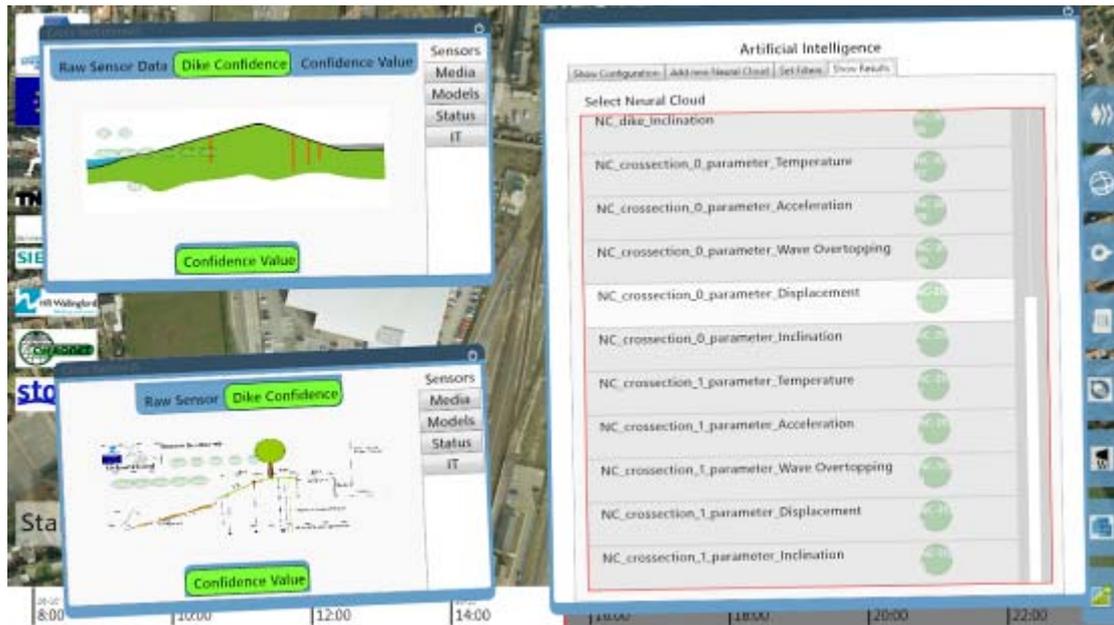


Figure 7 - Visualization of output of AI components of two running EWS

### 3.3 Currently running EWS

Thanks to current capabilities of the UrbanFlood distributed cloud platform and the CIS there are several objects under supervision of a monitoring EWS. They include:

- Live dike in Eemshaven (NL)
- Stammer dike in Amsterdam (NL)
- Zeeland dike (NL)
- Embankment in Boston (UK)

All these EWS have very similar structure with the AI component calculating the estimated dike confidence which triggers physical models by HR Wallingford when the confidence threshold is exceeded.

All data on these EWS can be seen in the UrbanFlood visualization as shown in Figure 8. For each monitored object it is possible to see the sensor locations in cross-sections, live and historical sensor readings, confidence values calculated by the AI components based on certain sets of the sensor data. Confidence values show the estimated confidence in the absence of anomalies in the object behaviour. These values are used to trigger more sophisticated and resource-intensive modelling appliances.

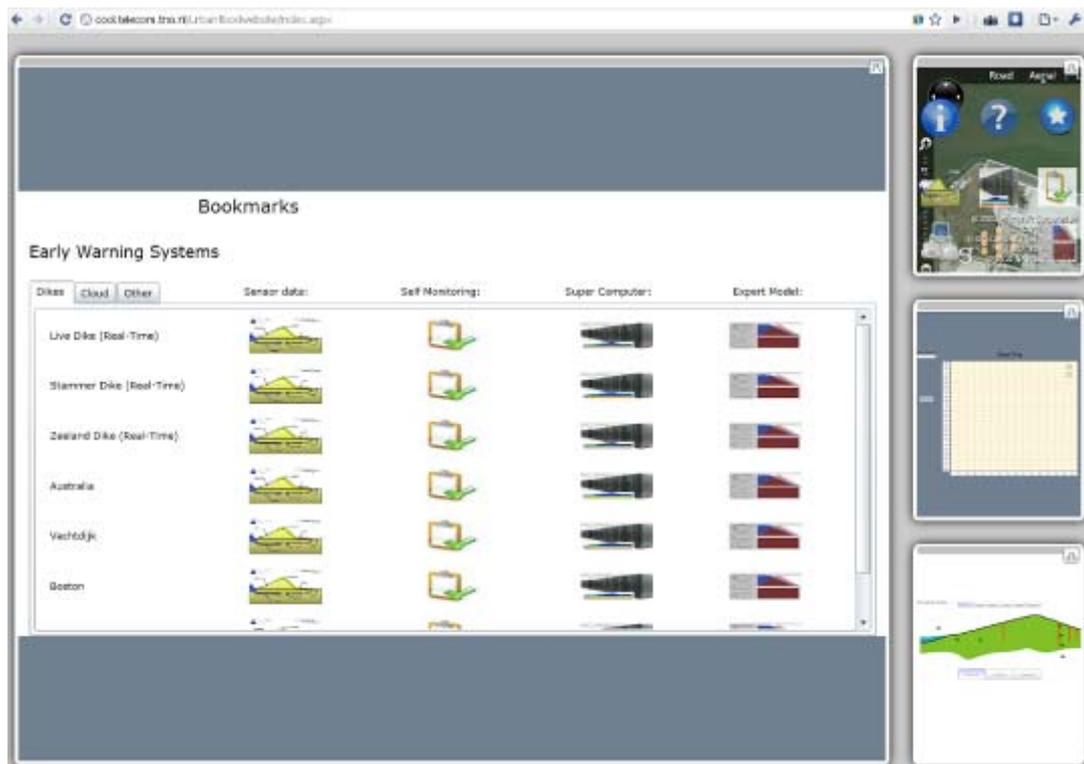


Figure 8 - Running UrbanFlood EWS

The Cloud infrastructure itself is also visible in the application. An overview of components used by the CIS is shown in Figure 9.



Figure 9 - Running CIS VMs

Separate views of specific VM's are possible as is shown in Figure 10. Here three machines running on a host in Poland are shown.

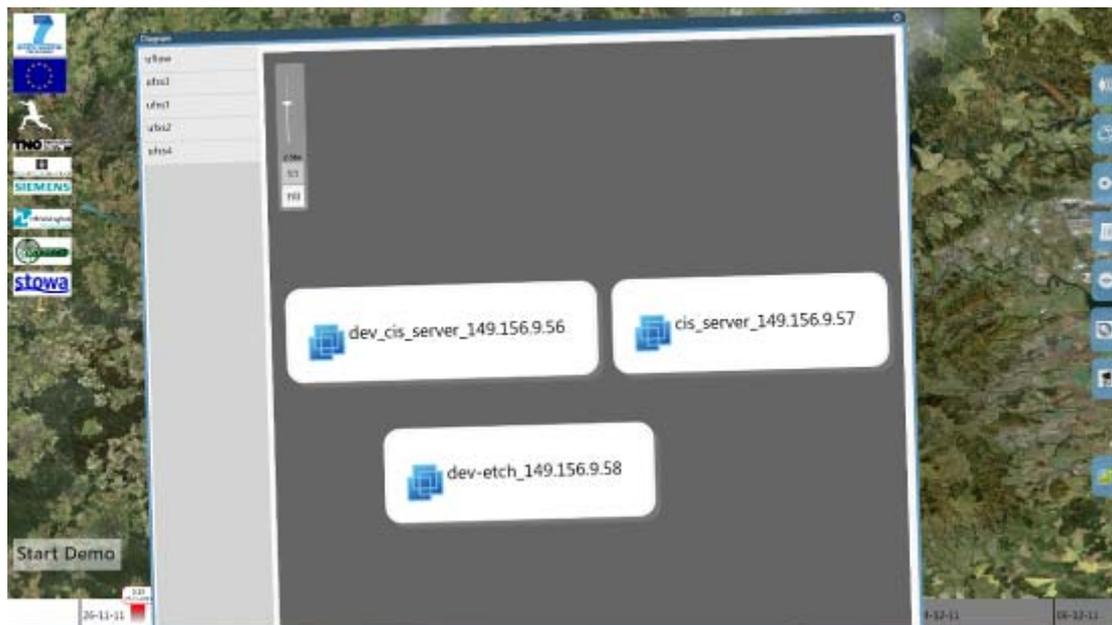


Figure 10 – Overview of Hosts and VMs running on a specific host

It is also possible to show usage graphs and statistics for each VM. This way the VMs can be monitored to allow warnings in case of overloading the EWS. Figure 11 shows a list of VM's and a graph of CPU usage. Included in the list is a visual representation of the different aspects of a VM such as CPU, Memory, Disk and Network. It is also possible to sort the list according to other parameters such as belonging to a specific EWS, or the Cost involved in running the VM.

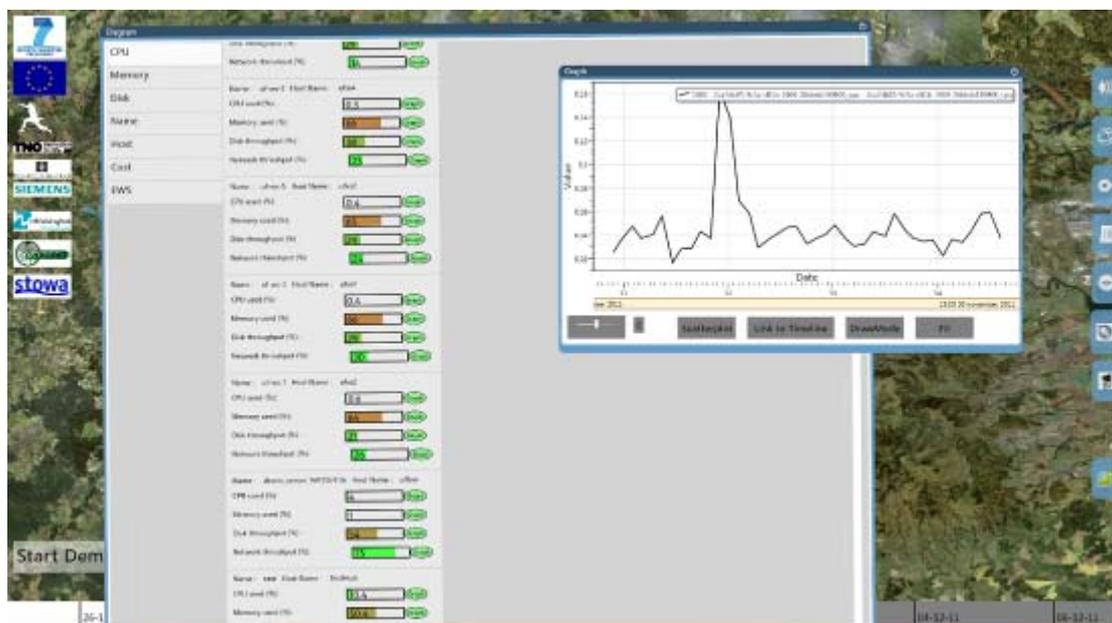


Figure 11 – Overview of VMs and usage statistics

## 4 References and bibliography

- [1] TNO (2009a). UrbanFlood proposal, Annex I – “Description of Work”
- [2] UrbanFlood Deliverable D5.2 - Specification of the architecture and interfaces of the Common Information Space
- [3] UrbanFlood Deliverable D6.2 – Operational Internet DSS viewer
- [4] UrbanFlood Deliverable D6.3 – Specification of Plug-in Technology for EWS Component