



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

OBSIDIAN 2.2 USER MANUAL

Obsidian 2.2

OBSIDIAN 2.2 USER MANUAL

© 2014-2021 Obsidian Soft
Published under brand authorization
First Edition: January 2017

Obsidian Soft
Pº General Martínez Campos 44 – 28010. Madrid
<https://www.obsidiansoft.com>
info@obsidiansoft.com

Authors: Raquel Fernández, Diego Berea
Designers: Severino Ribecca, Diego Berea



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE



TABLE OF CONTENTS

1

Introduction

2

Installation and Initial Setup

3

Service Modeling

4

Definition of Dashboards, Displays and Metrics

5

Configuration and Administration

6

Data Source Integration

7

Simulation Scenarios

8

Obsidian architecture



1

INTRODUCTION TO OBSIDIAN

Introduction

Platform features

Available Editions



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

OBSIDIAN USER MANUAL

This document contains the Obsidian 2.0 User Manual and is intended for Obsidian platform end-users.

The Obsidian administration guide is available in HTML format in Obsidian's technical wiki, along with the rest of Obsidian's technical documentation, which can be accessed from <http://wiki.obsidiansoft.com>

This manual refers to the Obsidian 2.0 Professional Edition.

All features from the Professional Edition are also included in the Enterprise edition.



OBSIDIAN CORE

Open source edition from the core of Obsidian, for developers and investigators.



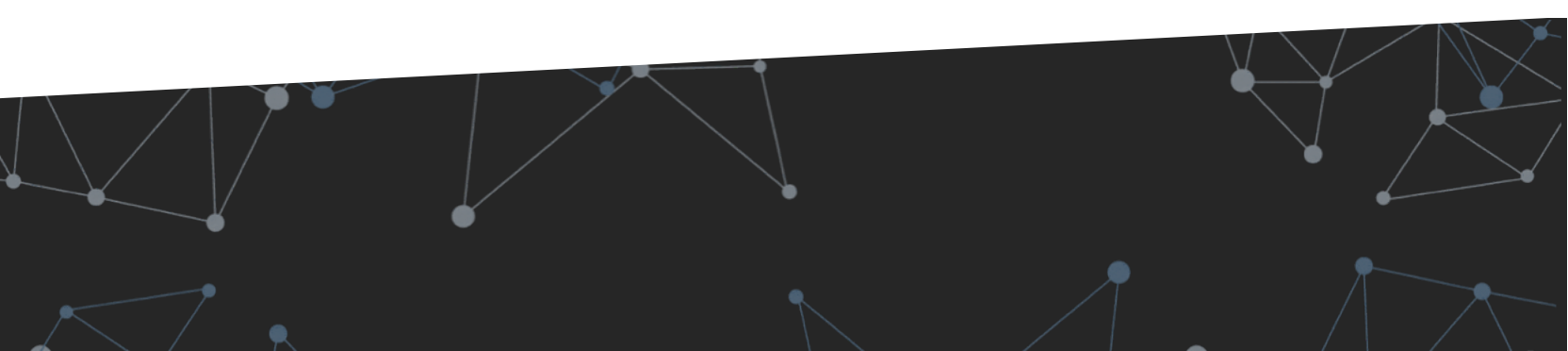
OBSIDIAN PROFESSIONAL

Commercial edition of Obsidian, ideal for any organization, with all platform features available.



OBSIDIAN ENTERPRISE

Advanced edition for distributed deployments, with parallel computing and other specialized requirements.



OBSIDIAN PLATFORM

Obsidian is a Business Service Intelligence & Analytics platform for the definition and automatic calculation of real-time and historical service level indicators for availability, capacity, service desk assistance and service level.

Through integration with the organization's pre-existing data sources (monitoring tools, service desk platforms, business appliances, platform management environments, etc.) it represents service architecture and dependencies.



ON-PREMISES & SaaS INSTALLATION

On-premises, SaaS or hybrid deployment available. Distributed architectures are also supported



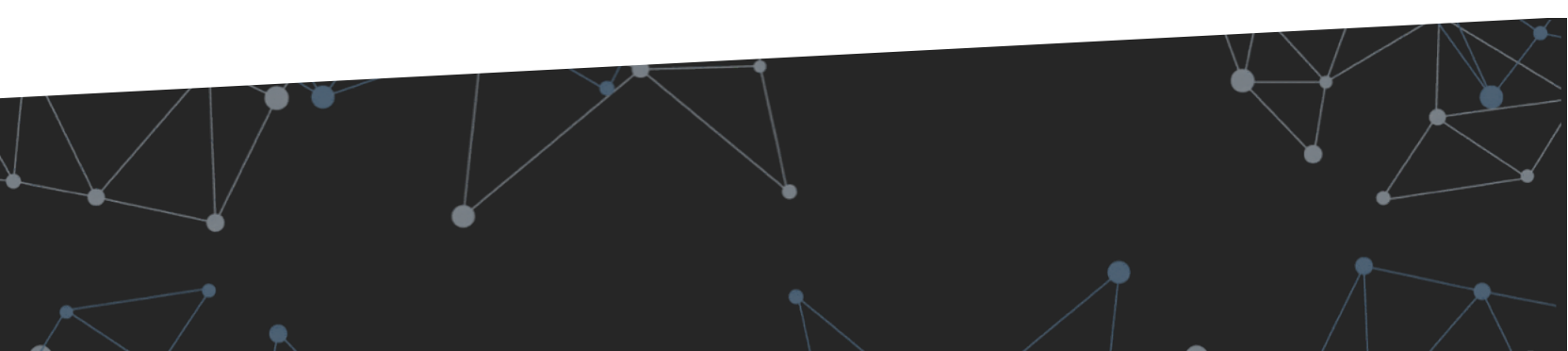
CORPORATE TOOL INTEGRATION

Easy integration with the organization's pre-existing monitoring and service desk tools



ITIL/ISO 20000 AND IT4IT SUPPORTED

Aligned with ITIL/ISO 20000 processes and IT4IT value flows



2

INSTALLATION AND INITIAL SETUP

Deployment options

Initial setup of Obsidian



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

DEPLOYMENT OF OBSIDIAN PROFESSIONAL

The deployment of Obsidian Professional is carried out via virtual GNU/Linux (openSUSE) appliances distributed in the clients datacentres. The appliances don't require internet connection to run.

Although Obsidian is usually deployed on-premises. Cloud platform deployments and hybrid on-premises/cloud platform deployments are also available.

The recommended approach is to deploy Obsidian “near” the data sources that will be used.

DOWNLOAD THE VIRTUAL APPLIANCE

Download the appliance from the following URL:

<http://rpm.obsidiansoft.com/appliance/>

It requires a PinSAFE username, which will be provided by your official Obsidian distributor.

INITIAL SETUP

1. Integrate appliance in the virtualization environment.
2. Start the appliance. The initial setup is automatic.
3. Open the appliance IP in a web browser to launch the setup wizard.

MINIMUM REQUIREMENTS

The minimum requirements for the virtual appliance are 2 cores and 4GB of RAM.

DEPLOYMENT IN ESX ENVIRONMENTS

Upon deploying virtual appliances in ESX environments, when the virtual environment's start-up process is finished, a message in green will appear with the IP address assigned to the appliance.

```
( 91.093876) Adjusting tsc more than 1%: (4730401 vs 6414663)
[ OK ] Stopped MySQL server.
      Starting MySQL server...
[ OK ] Started MySQL server.
[ OK ] Started Execute user-provided boot script, if it exists.
      Starting Execute the SUSE Studio first boot script, if it exists...
      Stopping wicked managed network interfaces...
[ OK ] Stopped wicked managed network interfaces.
      Starting wicked managed network interfaces...
( 97.867130) e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ OK ] Started wicked managed network interfaces.
[ OK ] Stopped Getty on tty1.
[ OK ] Created slice system-autologin.slice.
      Starting Getty on tty1...
[ OK ] Started Getty on tty1.
[ OK ] Stopped OpenSSH Daemon.
      Starting OpenSSH Daemon...
[ OK ] Started OpenSSH Daemon.
[ OK ] Started Getty on tty1.
( 115.193826) systemd[1]: Started Getty on tty1.
[ OK ] Reached target Login Prompts.
      Starting /etc/init.d/after.local Compatibility...
[ OK ] Started /etc/init.d/after.local Compatibility.
[ OK ] Reached target Multi-User System.

Welcome to openSUSE Leap 42.1 - Kernel 4.1.39-56-default (tty1).

obsidian login: obsidian (automatic login)
No mail.
Have a lot of fun...

Please access http://192.168.0.187 in your web browser

obsidian@obsidian:~$
```


DEPLOYMENT OF OBSIDIAN ENTERPRISE

Obsidian Enterprise incorporates a series of additional characteristics, further to those available in Obsidian Professional, suitable for environments that require a more advanced architecture.

VIRTUAL APPLIANCE DOWNLOAD

The download link will be provided by your Obsidian sales representative.

INITIAL SETUP

Setup is similar to Obsidian Professional setup:

- Integrate the appliance in the virtualization environment.
- Start the appliance. The initial setup is automatic
- Open the appliance IP in a web browser to launch the setup wizard.

MINIMUM REQUIREMENTS

The minimum requirements for virtual appliances are similar to the Professional version, 2 cores and 4GB of RAM.

However, due to the parallel calculation engine included in Obsidian Enterprise, increasing the number of available cores will reduce processing time.

The administration guide for Obsidian Enterprise includes analysis of Obsidian calculation parallel performance <https://wiki.obsidiansoft.com/enterprise/parallelization>.

RED HAT ENTERPRISE LINUX SERVERS

In addition to deploying Obsidian Enterprise via virtual GNU/Linux (openSUSE) appliances, it can also be deployed in Red Hat Enterprise Linux 7.x corporate servers provided by the client.

In this case, the Obsidian repository should be configured for RHEL environments:

1. Create `/etc/yum.repos.d/obsidian.repo` with the following content:

```
[obsidian]
name=obsidian
baseurl=http://rpm.obsidiansoft.com/redhat
enabled=1
gpgcheck=0
```

2. Update RPMs and install Obsidian packages:

```
# yum update -y
# yum install -y obsidian-base
obsidian-calculate obsidian-
dashboard obsidian-console obsidian-
extra obsidian-data2text
```

3. Start the initial Obsidian setup:

```
#/opt/ozona/utils/bin/setupappliance
```

INTERNET ACCESS AND APPLIANCE USERS

Obsidian doesn't require internet access to run. However, it may be useful to enable remote access or access to external URLs in certain circumstances:

ACCESS TO EXTERNAL URLS FROM THE OBSIDIAN APPLIANCE

We recommend you enable access to standard openSUSE and Obsidian repositories (<http://rpm.obsidiansoft.com>) for frequent appliance updates.

REMOTE ACCESS TO THE OBSIDIAN APPLIANCE

Your official Obsidian distributor can request remote access to the appliance via SSH and HTTP in order to streamline installation and configuration tasks.

APPLIANCE USERS

Obsidian's presentation layer is an Apache environment, meaning any type of authentication compatible with Apache will be valid: integration with active directory or LDAP, PinSAFE double factor security authentication, etc.

Obsidian provides a series of default local users.

DEFAULT USERS

User: obsidian (Spanish)

Password: To be requested

User: obsidianEN (English)

Password : To be requested

User: obsidianPT (Portuguese)

Password: To be requested

User: root

Password: To be requested

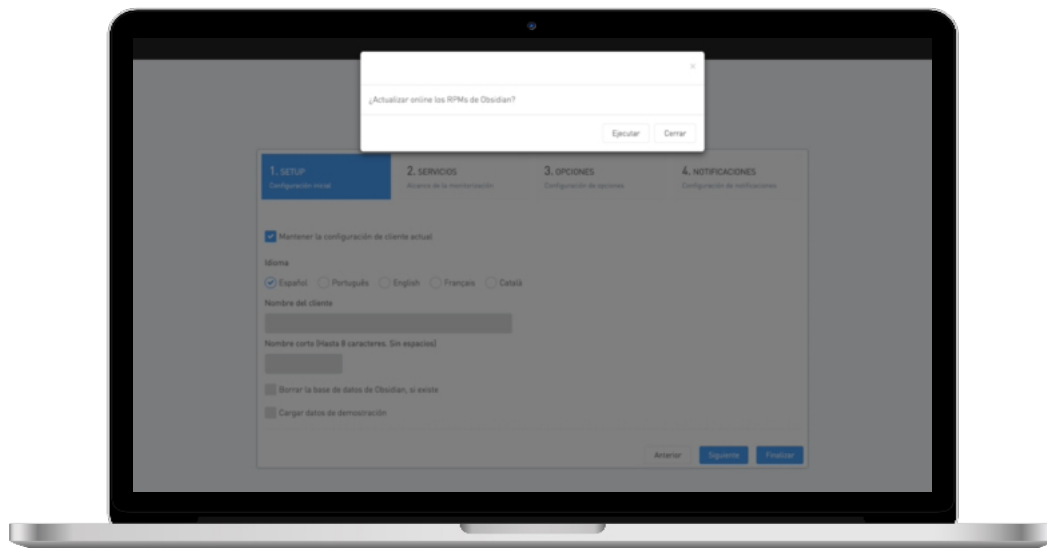
We recommend disabling the SSH as root in environments published on the Internet.

APPLIANCE CONFIGURATION (I)

When logging in for the first time to the web appliance, the initial setup wizard will be automatically launched.

If the appliance is online, it will request an update of the Obsidian RPM packages.

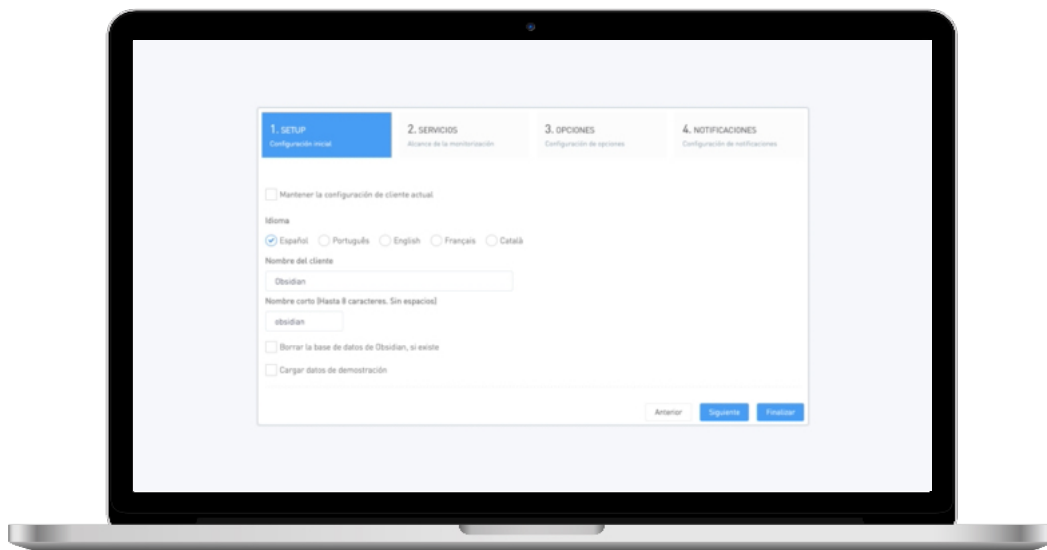
We recommend you update by selecting “Run”.



Choose the language and client name.

NOTE

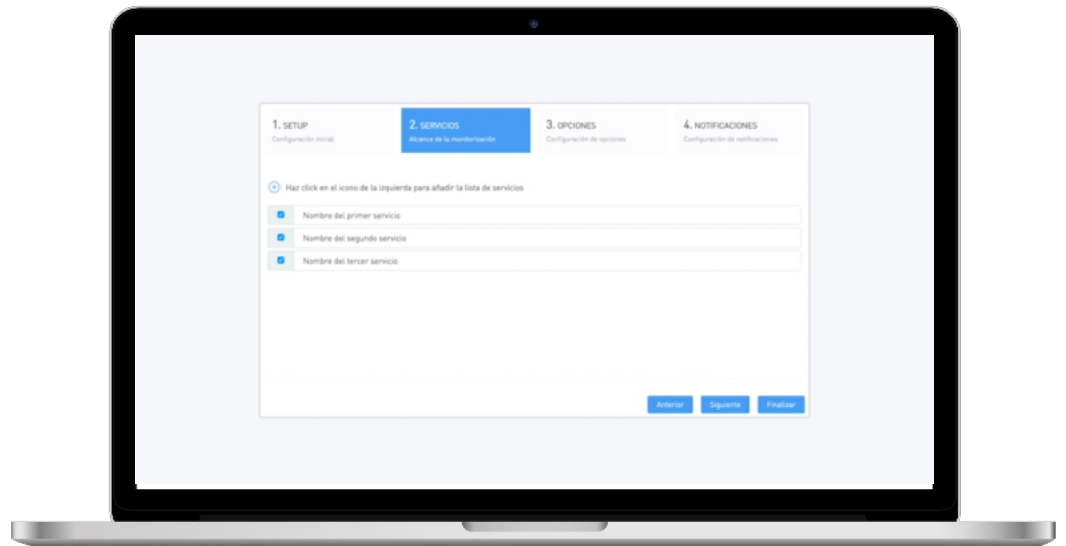
For concept tests, you can load dummy data to immediately finish the setup.



APPLIANCE CONFIGURATION (II)

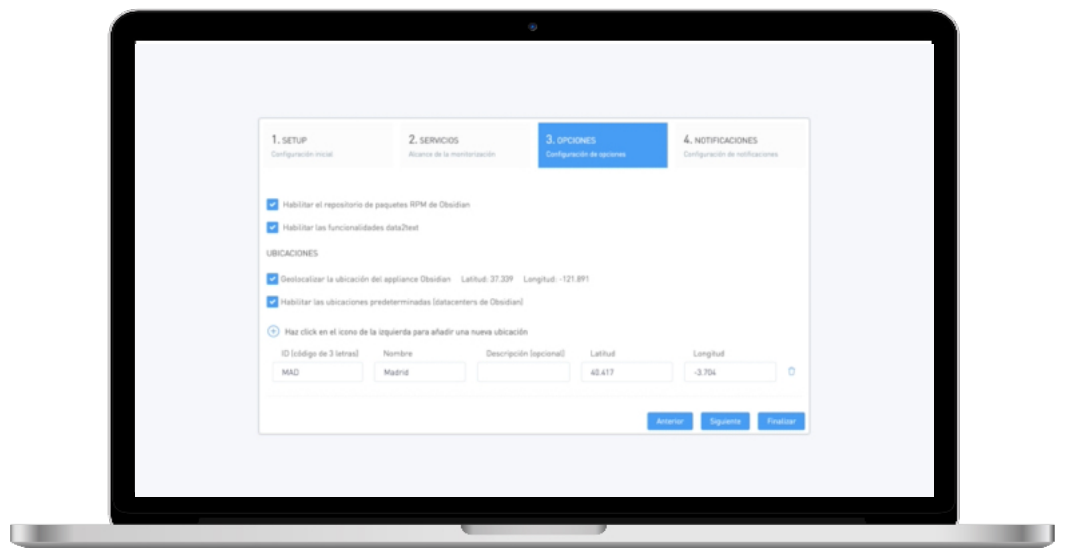
Add the services to be included in the scope.

You will be able to revisit this at a later date and add more services.



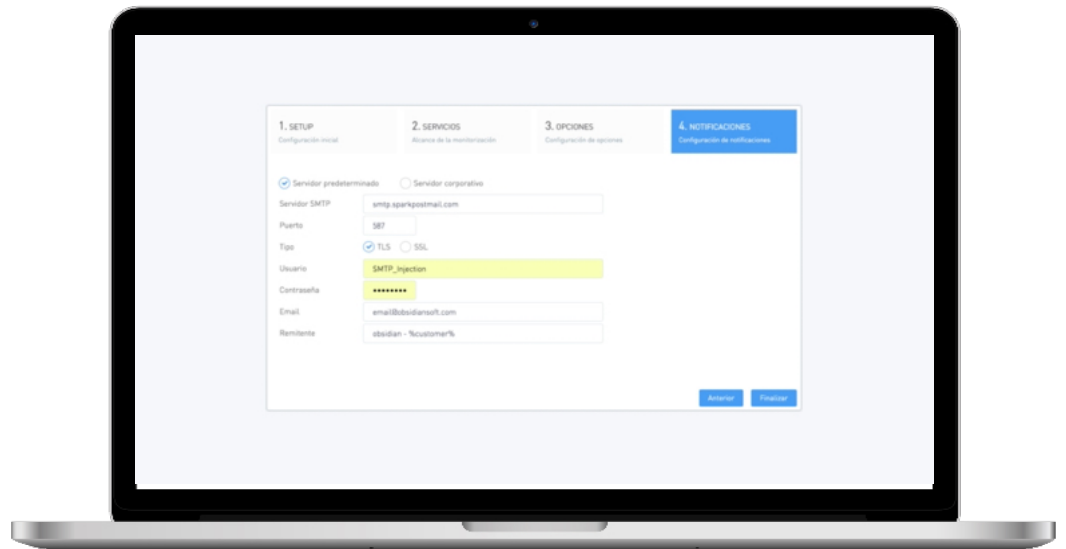
Add the location of datacentres and offices from where transactions will be launched.

You will be able to revisit this at a later date and add more locations.

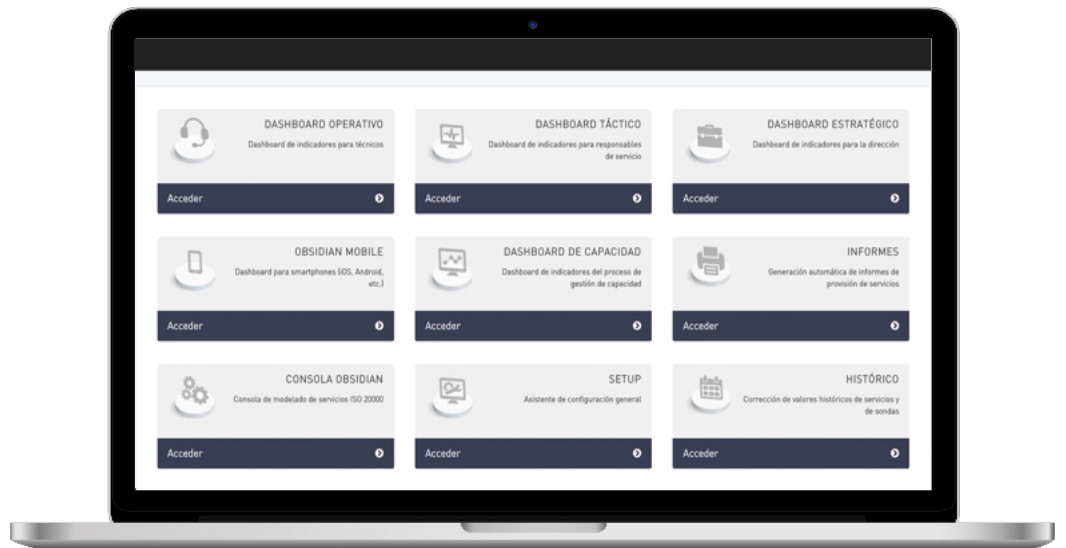


APPLIANCE CONFIGURATION (III)

Configure the current email server for notifications

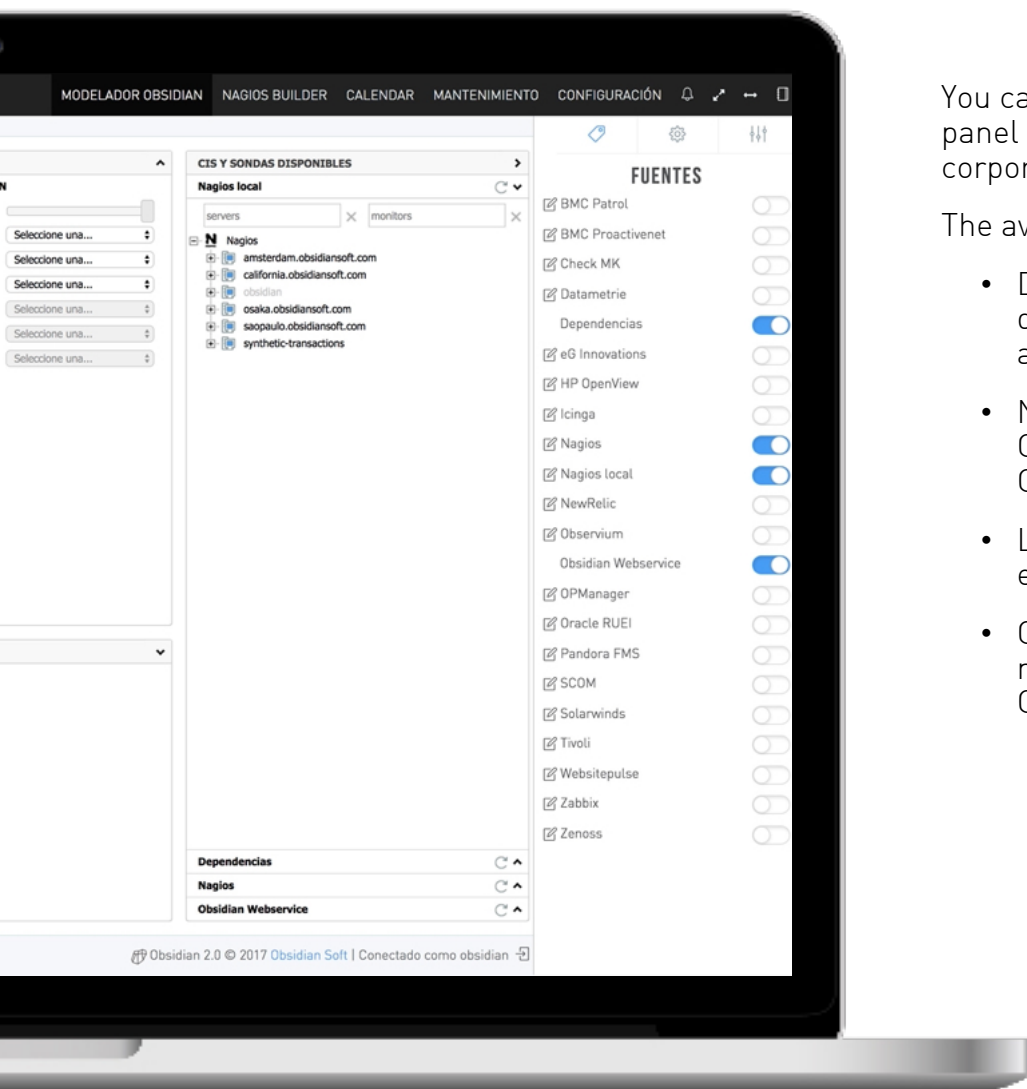


Dashboards and default reports are made immediately available by accessing the appliance IP via a web browser.



APPLIANCE CONFIGURATION (IV)

Upon finishing the wizard, you will be automatically redirected to the modeling console, where integrations with corporate data sources can be configured. To do this you need to access the side panel by clicking on the last icon of the top black toolbar.



You can then access the console's side panel and configure integration with corporate monitoring tools.

The available default tools are:

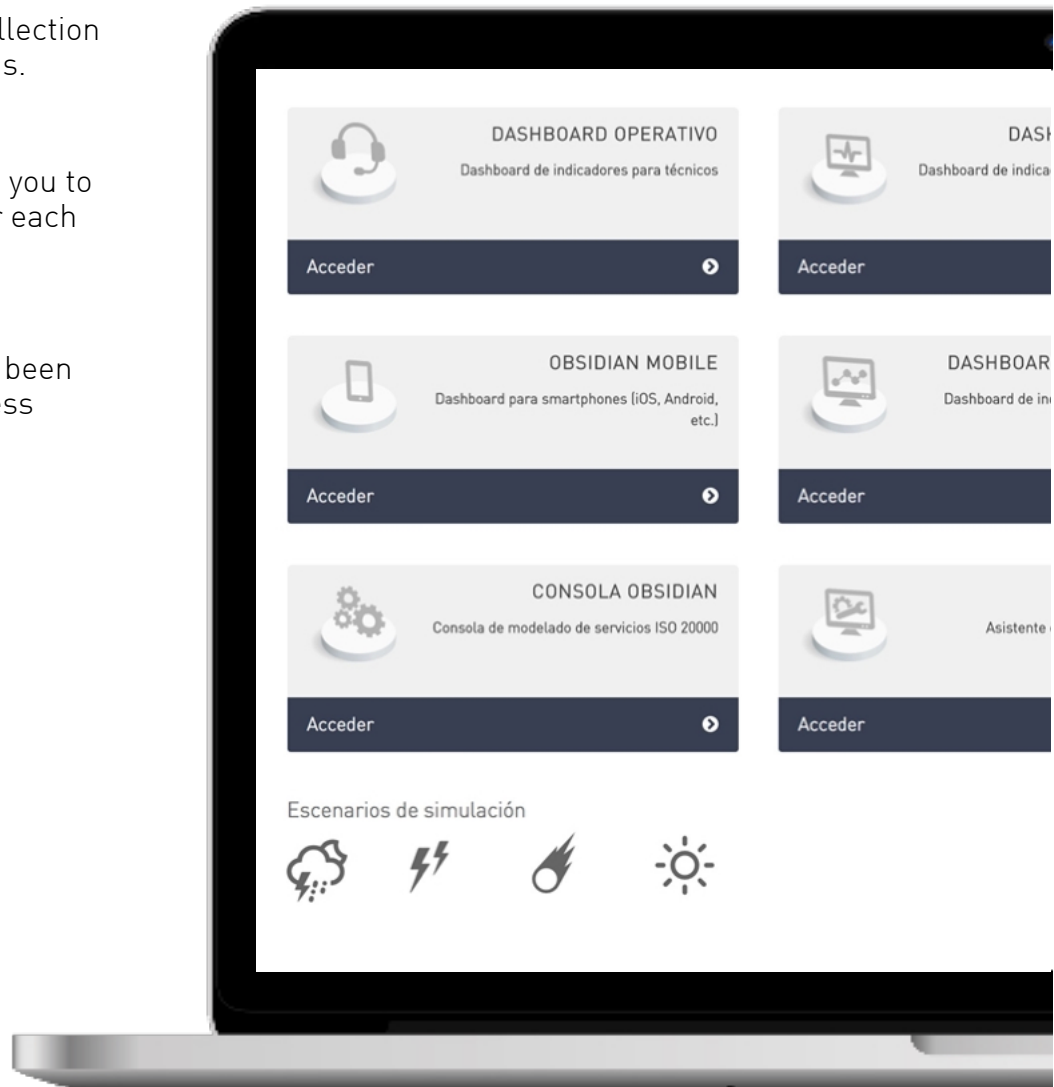
- **Dependencies:** To establish dependencies between one service and another.
- **Nagios:** Requests for remote Nagios. Configured by default for access to an Obsidian demo server.
- **Local Nagios:** Default Nagios environment available in the appliance.
- **Obsidian webservice:** For modeling metrics and indicators sent to Obsidian's REST webservice.

OBSIDIAN MAIN MENU

The Obsidian main menu offers a collection of the most common shortcut options.

This menu is customizable, allowing you to display the most relevant options for each client.

If simulation scenarios have already been defined, you will also be able to access them from this menu.



3

SERVICE MODELING

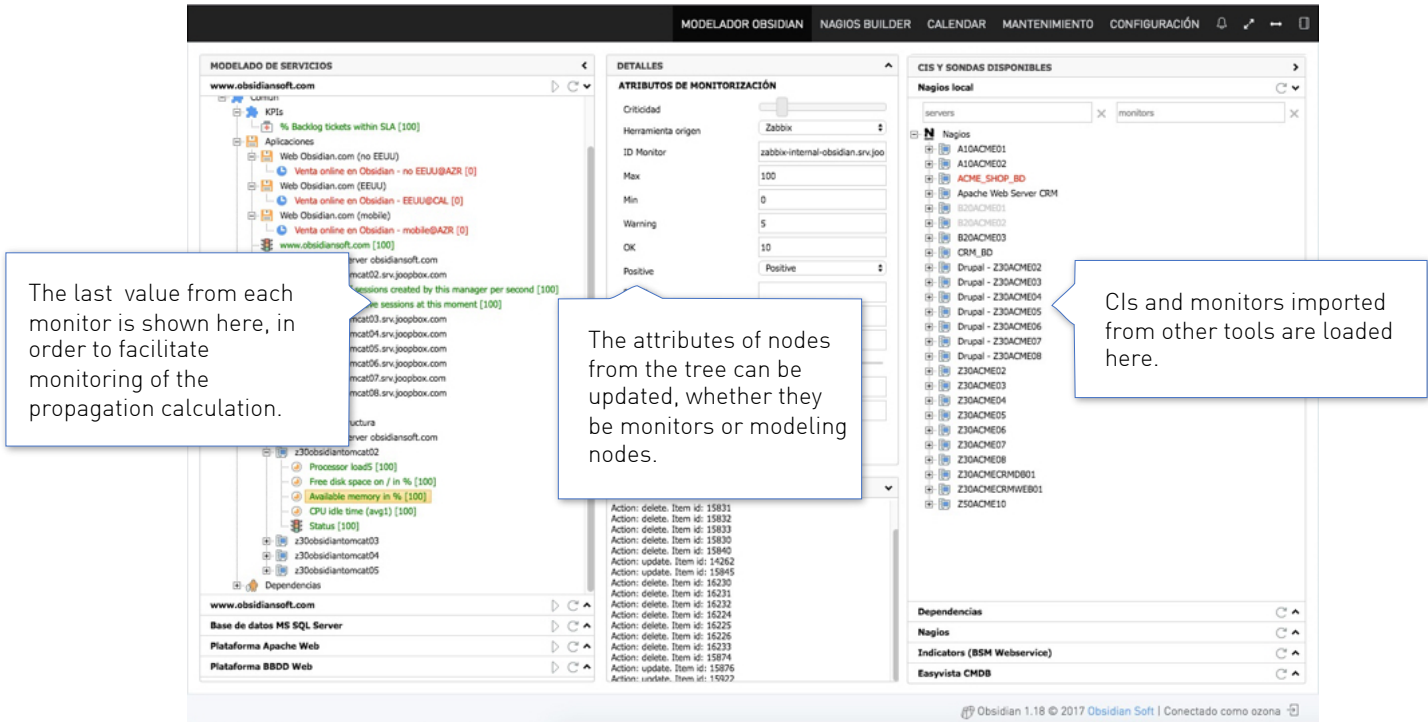


OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

SERVICE MODELING CONSOLE

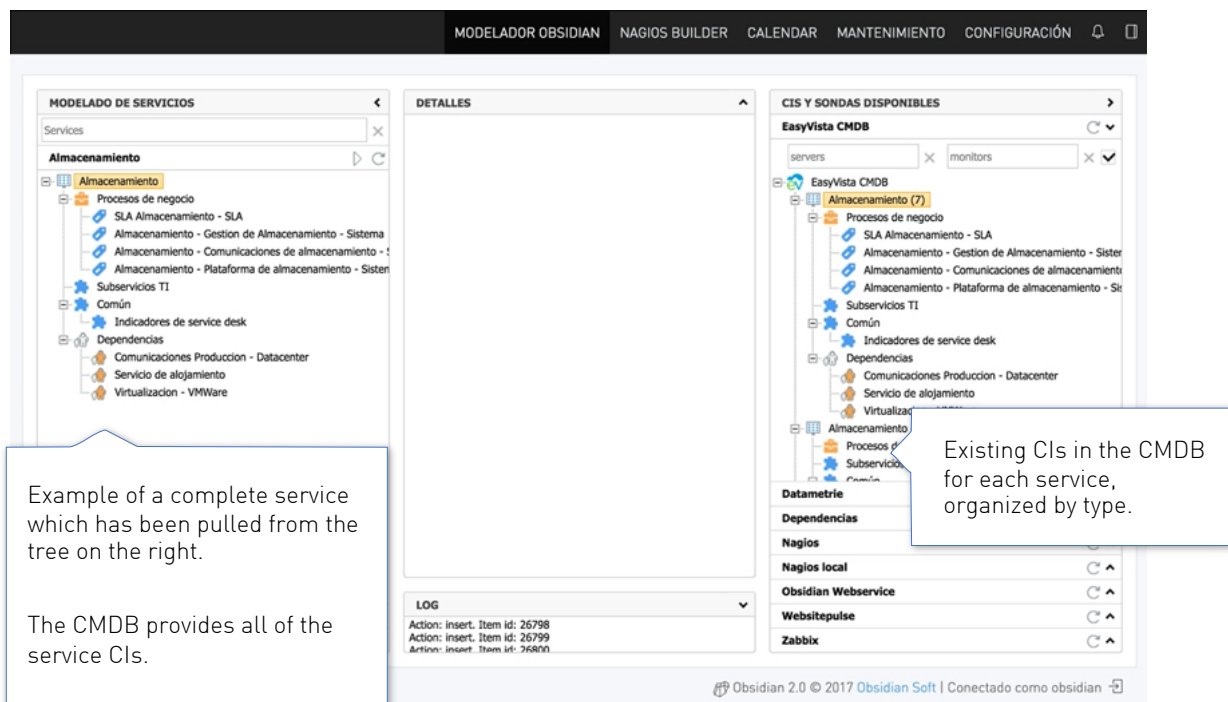
KEY FEATURES OF THE SERVICE MODELING CONSOLE

The modeling console consists of three columns: service modeling, characteristics and data sources



INTEGRATION OF THE MODELING CONSOLE WITH CMDB TOOLS

A CMDB provides service architecture and simplifies the definition and maintenance of modeling.

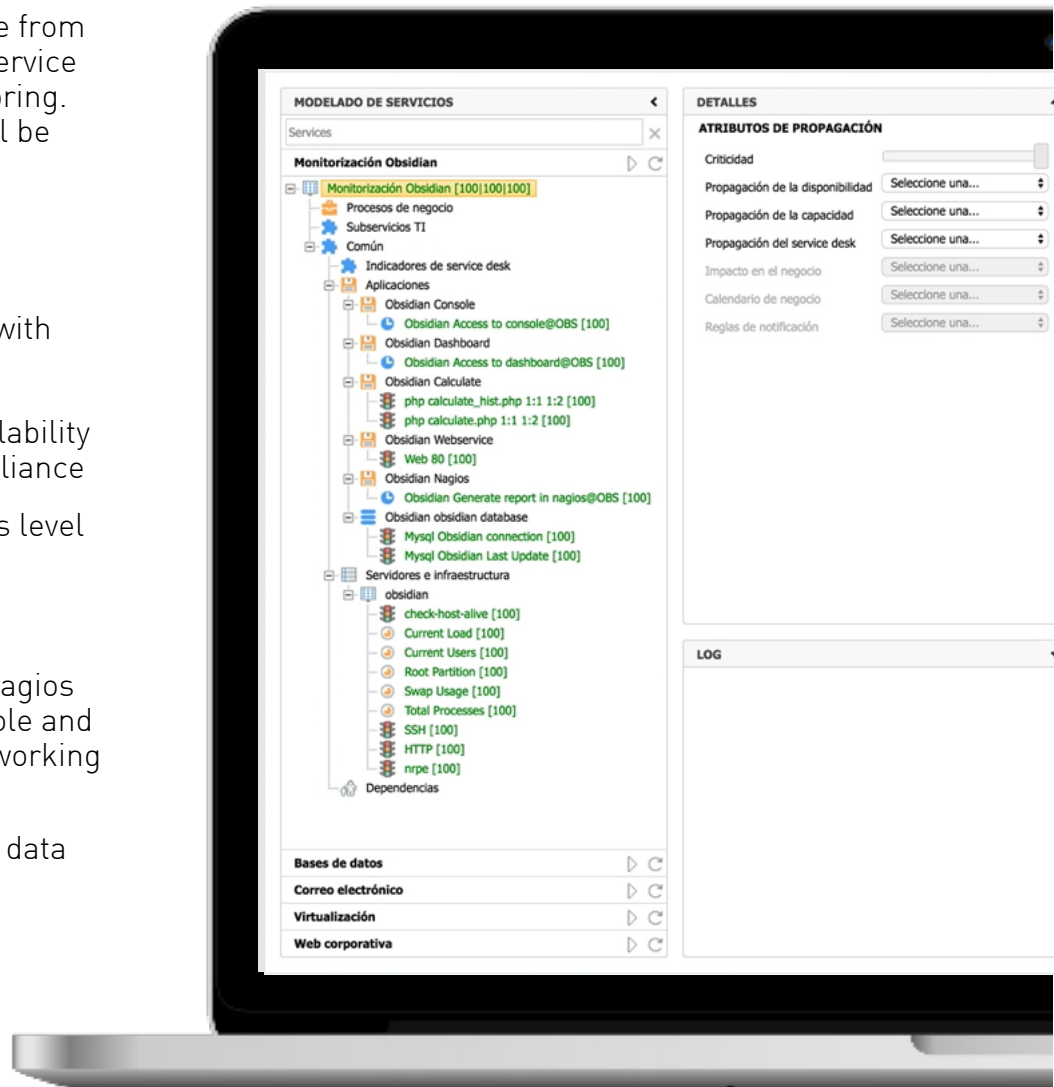


EXAMPLE SERVICE: “OBSIDIAN MONITORING”

After running the initial wizard, aside from the defined services, an additional service is created for Obsidian's own monitoring. This is an example service which will be used in this manual to illustrate the different configuration options.

It is a service comprised of a single machine, Obsidian's own appliance with different types of monitors:

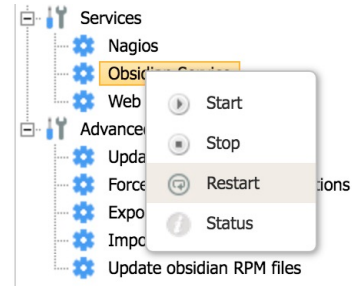
- At infrastructure level, with availability and capacity metrics for the appliance
- At application level, with process level monitors and database instance monitors.
- User experience, with synthetic transactions to check that the Nagios local server, the modeling console and the Obsidian dashboard are all working correctly
- Service desk indicators (dummy data for demo purposes)



EXAMPLE SERVICE: "OBSIDIAN MONITORING"

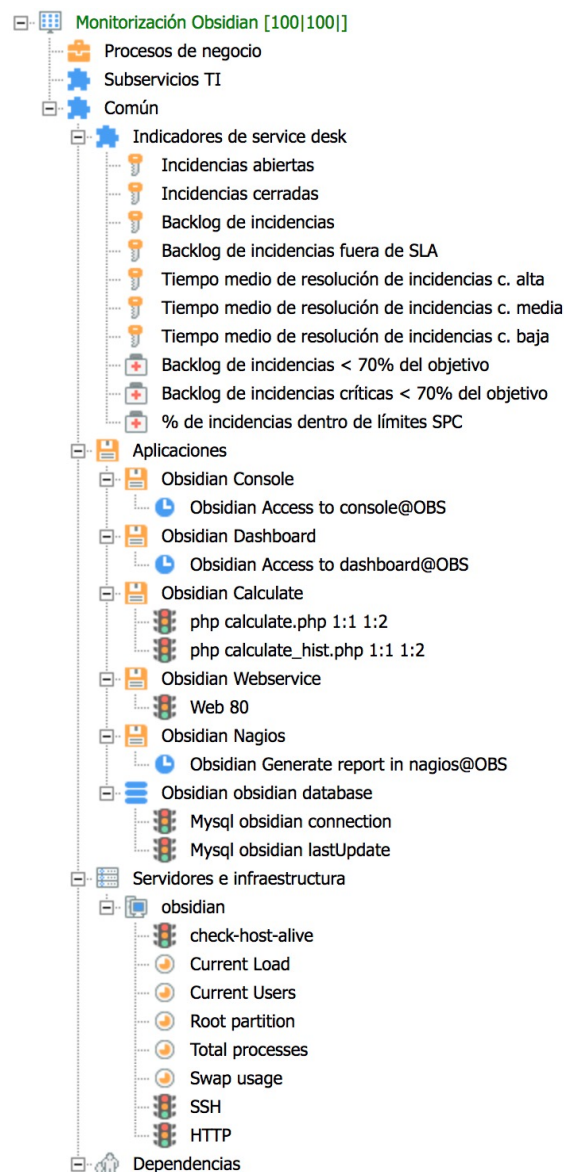
Once services have been modeled, the Obsidian calculation engine should be restarted by selecting the following options:

Configuration → Operation → Obsidian Service → Restart



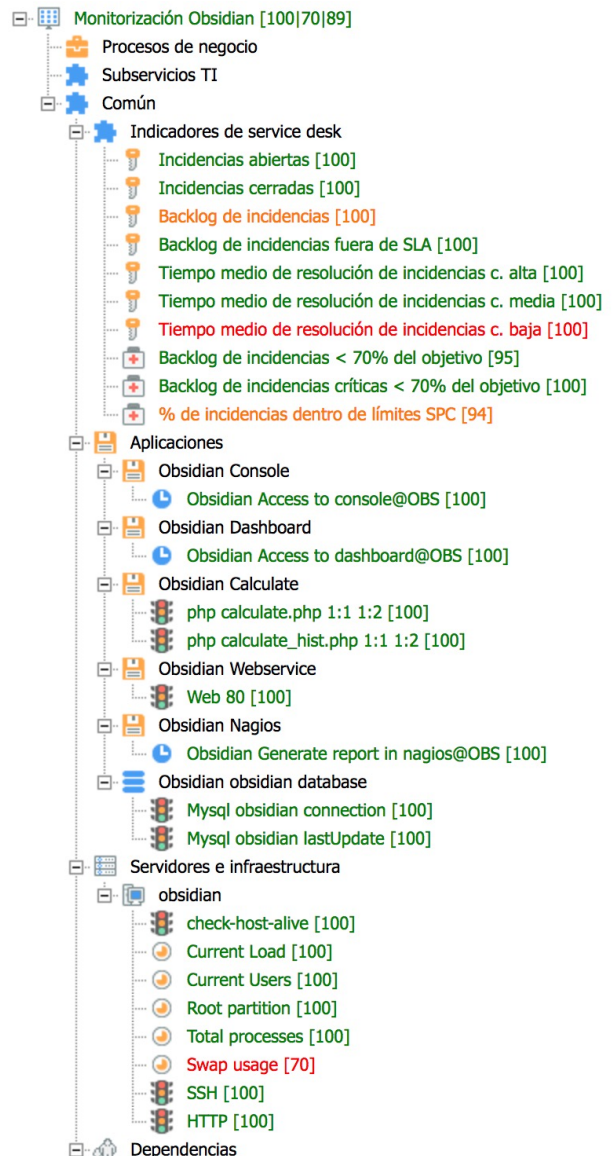
BEFORE STARTING THE CALCULATION MOTOR

The monitors (leaf nodes) have no associated values



AFTER STARTING THE CALCULATION MOTOR

The monitors obtain values from data sources



SERVICE MODELING CONSOLE

CIS AND MONITORS PROVIDED BY MONITORING TOOLS

Example of the integration of CIs and monitors coming from monitoring tools with the model.

The screenshot displays the 'MODELADOR OBSIDIAN' interface with three main panels: 'MODELADO DE SERVICIOS', 'DETALLES', and 'CIS Y SONDAS DISPONIBLES'. The 'MODELADO DE SERVICIOS' panel shows a tree view of services like 'Internet' and 'ACME_SHOP_BD'. The 'DETALLES' panel shows configuration for a monitor, including 'Herramienta origen' (Nagios) and 'ID Monitor' (localhost+2016). The 'CIS Y SONDAS DISPONIBLES' panel lists available CIs and monitors, with some highlighted in grey. Three callout boxes provide context: 1. 'Example of a CI and its associated monitors which has been pulled from the tree on the right. The configuration of monitors is automatic.' 2. 'Indicators show if the webservice has any recent values, values from more than two days ago or has no monitoring values.' 3. 'CIs and monitors used in more than one service. For design reasons, it should be avoided.' 4. 'Available CIs and monitors that have been used previously in a service model are shown in grey.'

INDICATORS PROVIDED BY SERVICE DESK TOOLS

Example of the integration of indicators from service desk tools with the model.

The screenshot displays the 'MODELADOR OBSIDIAN' interface with three main panels: 'MODELADO DE SERVICIOS', 'DETALLES', and 'CIS Y SONDAS DISPONIBLES'. The 'MODELADO DE SERVICIOS' panel shows a tree view of services like 'Monitorización Obsidian' and 'Indicadores de service desk'. The 'DETALLES' panel shows configuration for a monitor, including 'Herramienta origen' (Obsidian Webservice) and 'ID Monitor' (SO-INCL3#99). The 'CIS Y SONDAS DISPONIBLES' panel lists available indicators from the service desk tool, with some highlighted in grey. Three callout boxes provide context: 1. 'Example of a set of service desk KPIs that has been pulled from the tree on the right. Configuration of the KPIs is automatic.' 2. 'Indicators show if KPIs have recent values, values from more than two days ago or have no values, for the service being modeled.' 3. 'Service desk indicators provided by the service desk tool for the webservice.'

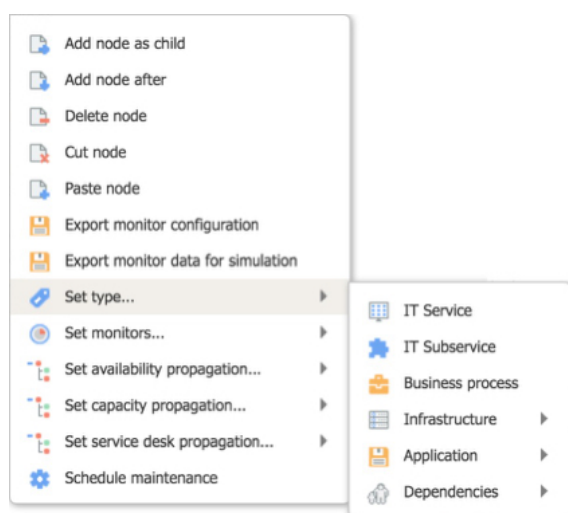
TYPES OF NODES

The types of nodes available in the model are the following:

- IT Service
- IT Subservice
- Business process
- Infrastructure element
- Application
- Dependency

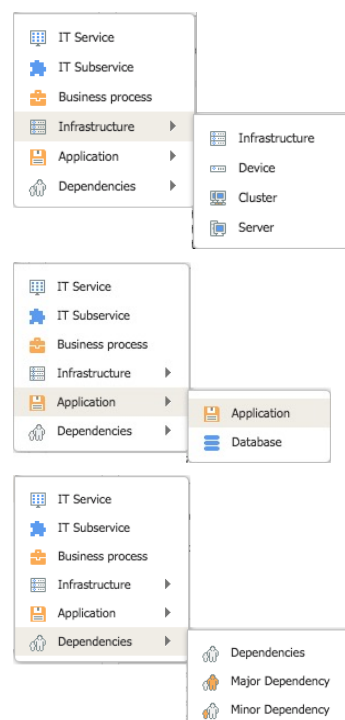
The first three types allow aggregated indicators and impact rules to be shown to different levels of granularity, using monitors associated with the remaining types of nodes or their own monitors.

All nodes represented in the model must correspond to CIs from the service architecture which are relevant to the aggregate calculation of availability, capacity or service level services.



The standard node subtypes available in the model are the following:

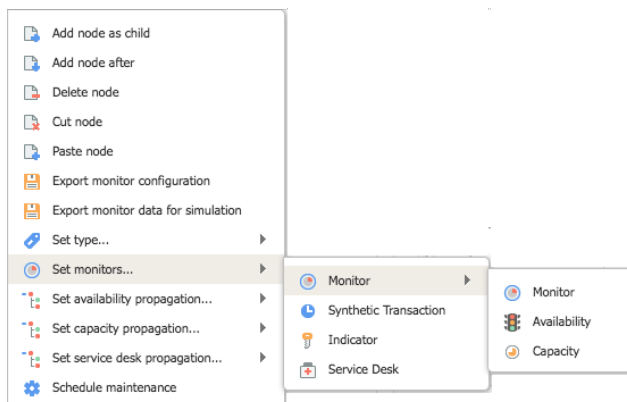
- Infrastructure element
 - Generic infrastructure element
 - Device
 - Cluster
 - Server
- Application
 - Application
 - Database
- Dependency
 - Dependency with no impact
 - Major dependency
 - Minor dependency



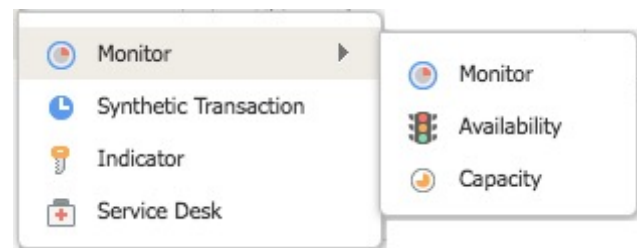
TYPES OF MONITORS

Any of the model nodes can be associated to the following types of monitors:

- Monitor
 - Generic monitor, undefined
 - Availability monitor
 - Capacity monitor
- Synthetic transaction
- Generic indicator
- Service desk indicator



Since ISO 20000 processes are used as a base, the model focuses on the calculation of aggregated availability and aggregated capacity for each service.



Analysis should be done to determine whether other dimensions of aggregated analysis should be considered:

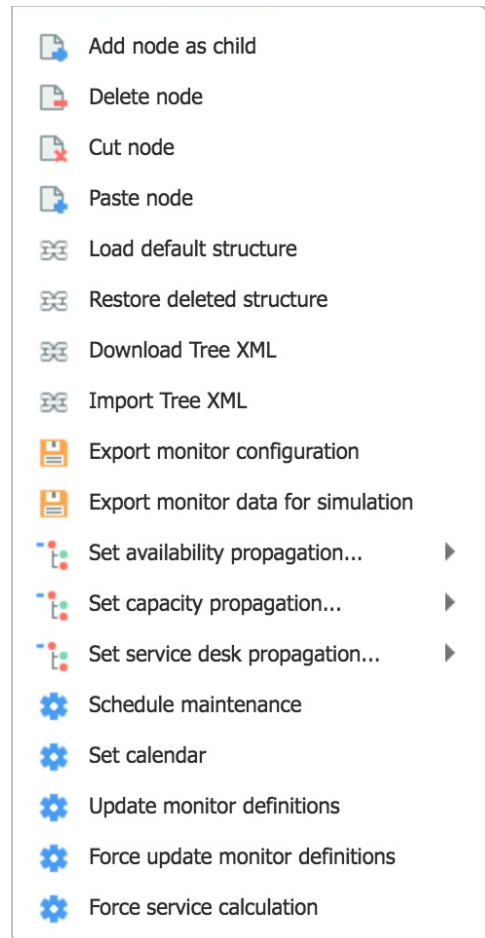
- Performance
- Information security
- Risk
- Cost
- Business impact
- Etc.

At present, those other dimensions may be included as part of the analysis, provided that they are linked to the availability or capacity calculation.

AVAILABLE FEATURES FOR SERVICE MODELING

Accessible via the shortcut node menu:

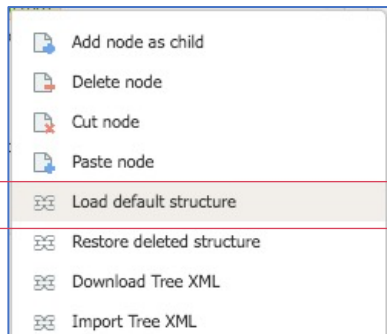
- Add node as child
- Add node
- Delete node
- Cut node
- Paste node
- Load default IT service structure
- Restore default IT service structure
- Download Tree XML
- Import Tree XML
- Export monitor configuration
- Export monitor data for simulation
- Set availability propagation criteria
- Set capacity propagation criteria
- Set service desk propagation criteria
- Schedule maintenance/scheduled service break
- Set calendar
- Update monitor definitions
- Force update monitor definitions
- Force service calculation
- Drag & drop between nodes of the same tree
- Drag & drop between nodes of different trees
- Implementation of changes to all elements of a multiple selection.



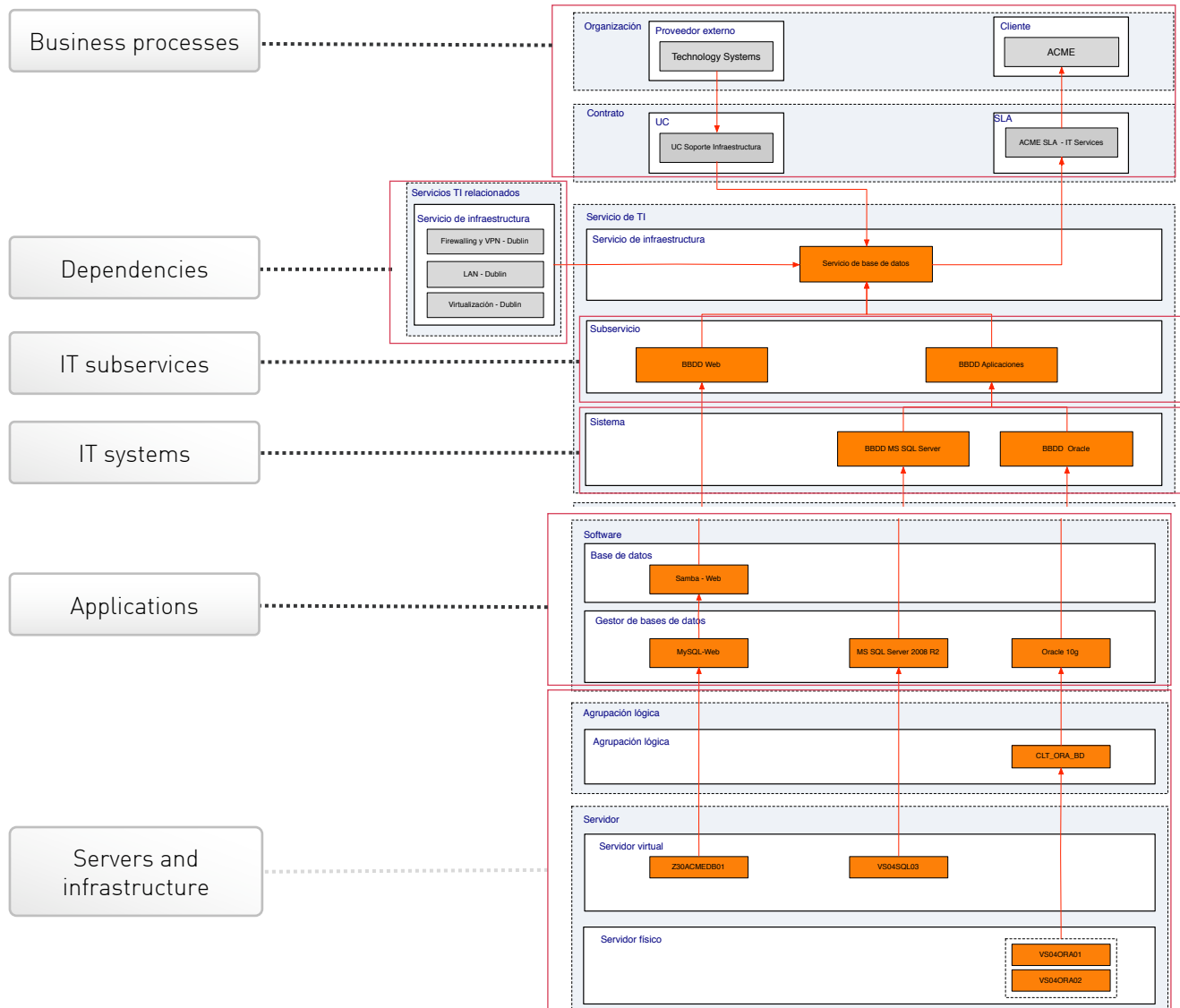
MODELING CRITERIA

DEFAULT SERVICE STRUCTURE

The default service structure can be loaded via the shortcut:



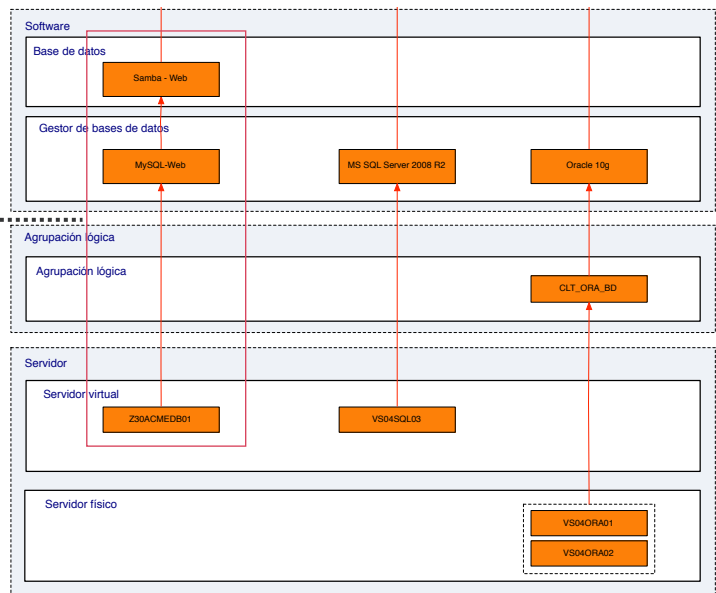
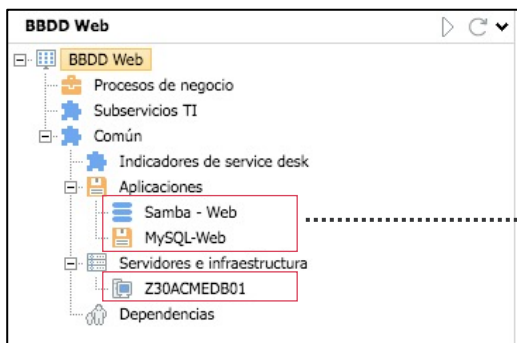
The default structure is made up of the following nodes in the modeling tree:



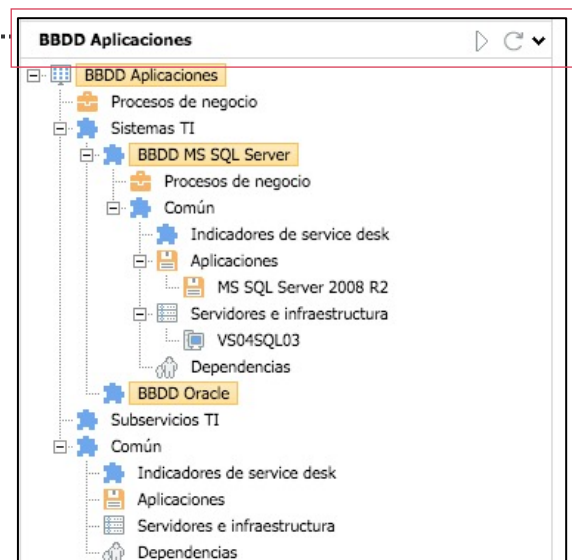
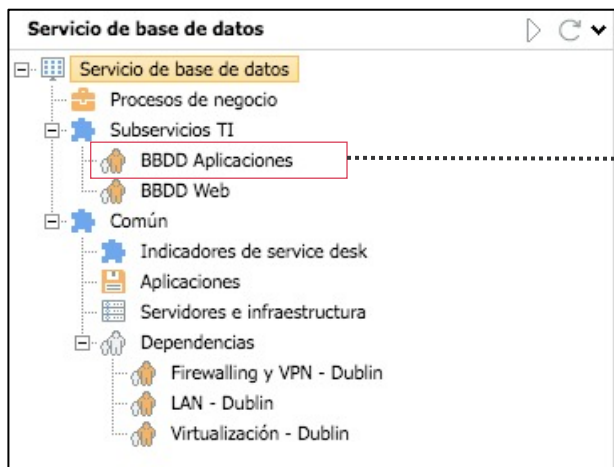
MODELING CRITERIA

RECOMMENDATIONS

- All model nodes –which are not monitors- should be included in the service diagram.
- The model does not need to reflect dependencies as “parent-child” relationships between applications and servers that contain them, as it is not relevant for the calculation.



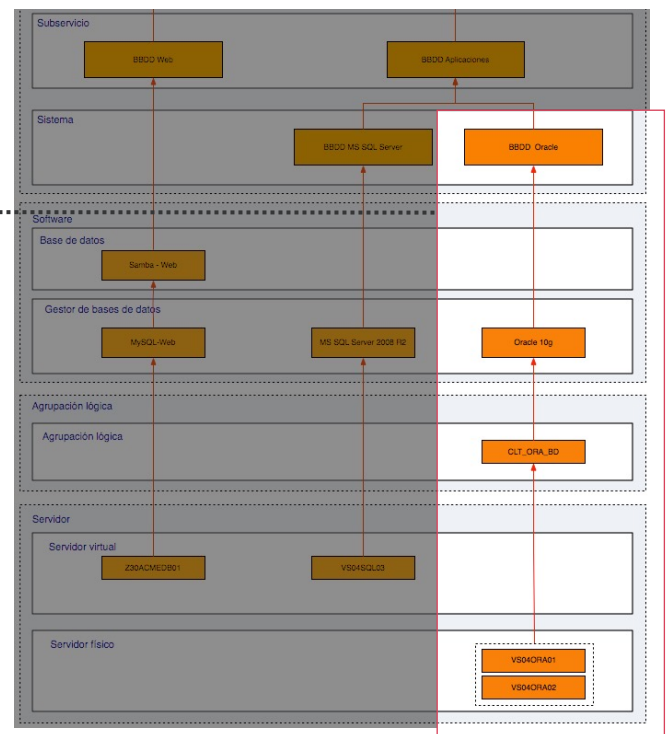
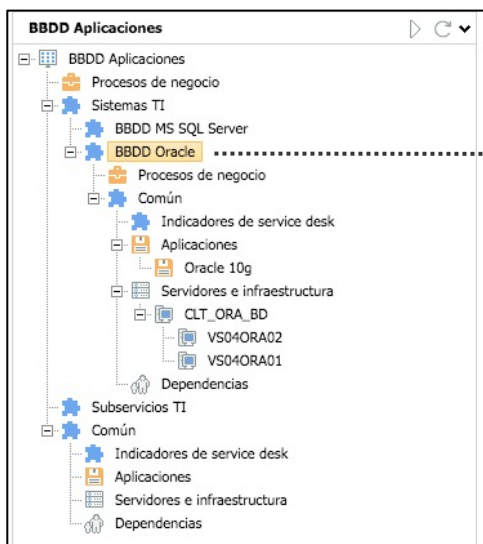
- The elements identified in the service diagram for each 'Subservice' will be modeled as components of an independent service



MODELING CRITERIA

RECOMMENDATIONS

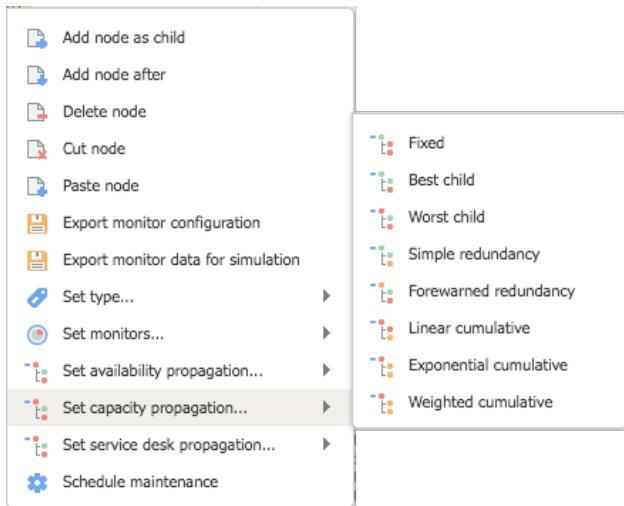
- The elements identified in the service diagram for each 'System' will be modeled within the 'IT Systems' node. In this way, this node will become a model sub-tree within the main service, with its own default modeling structure.



- Both physical services and virtual services will be modeled within the "Servers and infrastructure" node without considering its type.
- In the "Dependencies" node, all services that impact the service model in a major or minor way will be pulled.
- Only IT Services and IT Subservices identified in the service diagrams will be modeled.

PROPAGATION OF CRITICALITY RULES

For both availability and capacity, each node may have a different propagation of criticality rule.

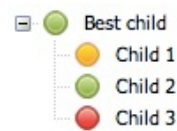


DEFAULT PROPAGATION RULES

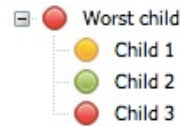
- Fixed: propagates a fixed value, without considering its child components



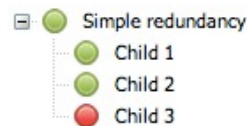
- Best child: propagates the value of the best child



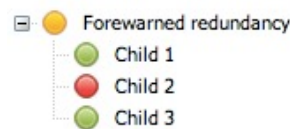
- Worst child: propagates the value of the worst child



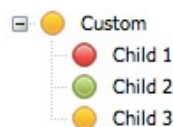
- Simple redundancy: returns the best value from the inner nodes



- Forewarned redundancy: returns the best value from the inner nodes, but lowering its value



- Linear cumulative
- Exponential cumulative
- Weighted cumulative



4

DEFINING DASHBOARDS, DISPLAYS AND METRICS



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

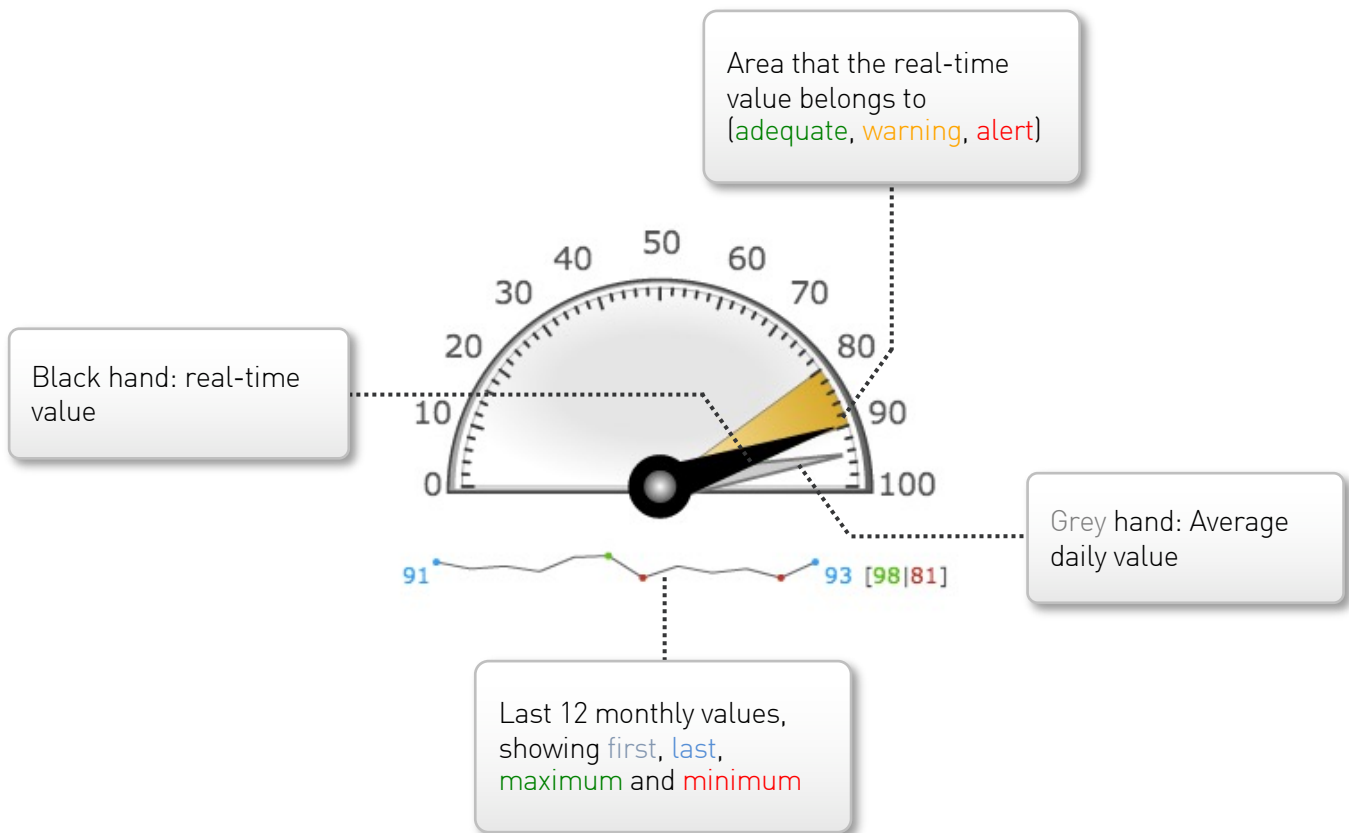
MAIN SERVICE INDICATORS

For each service, a real-time calculation of the most important aggregated indicators is provided:

- Service availability
- Service capacity
- Quality of attention received by users (service desk)
- And, as a calculation of the above, numeric quantification of service level.

Four values are shown for each indicator:

- Real-time value and area it corresponds to
- Average daily value
- Trend line
 - Last 12 monthly values (default)
 - Last 30 daily values
 - Last 24 daily values



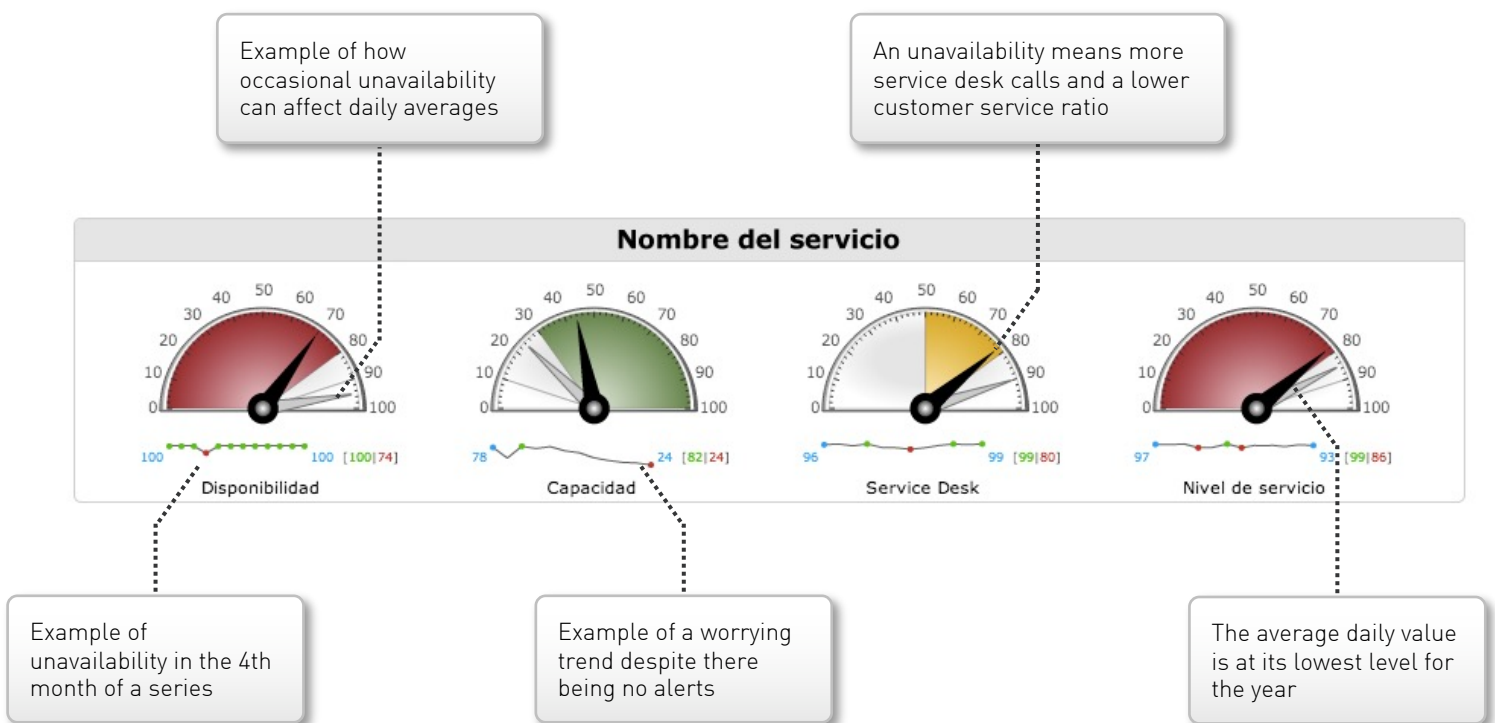
MAIN SERVICE INDICATORS

For each service, a real-time calculation of the most important aggregated indicators is provided:

- Service availability
- Service capacity
- Quality of attention received by users (service desk)
- And, as a calculation of the above, numeric quantification of service level.

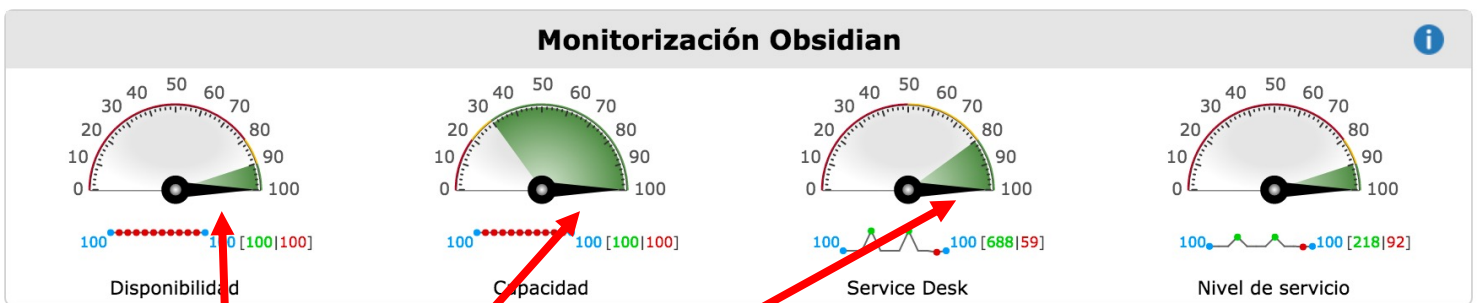
Four values are shown for each indicator:

- Real-time value and area it corresponds to
- Average daily value
- Trend line
 - Last 12 monthly values (default)
 - Last 30 daily values
 - Last 24 daily values



MODEL CORRESPONDENCE → DASHBOARD INDICATORS

The values calculated in the model tree at node root level, correspond to service level aggregated indicators.



The screenshot shows the "MODELADO DE SERVICIOS" interface. The "Monitorización Obsidian" node is selected in the model tree, and its details are shown in the "DETALLES" panel. The "CIS Y SONDAS DISPONIBLES" panel shows the available Nagios monitors.

MODELADO DE SERVICIOS

- Services
- Monitorización Obsidian [100|100|100]
- Procesos de negocio
- Subservicios TI
- Común
- Indicadores de service desk
- Aplicaciones
- Obsidian Console
- Obsidian Access to console@OBS [100]
- Obsidian Dashboard
- Obsidian Access to dashboard@OBS [100]
- Obsidian Calculate
- php calculate_hist.php 1:1 1:2 [100]
- php calculate.php 1:1 1:2 [100]

DETALLES

ATRIBUTOS DE PROPAGACIÓN

- Criticidad
- Propagación de la disponibilidad: Seleccione una...
- Propagación de la capacidad: Seleccione una...
- Propagación del service desk: Seleccione una...
- Impacto en el negocio: Seleccione una...
- Calendario de negocio: Seleccione una...
- Reglas de notificación: Seleccione una...

CIS Y SONDAS DISPONIBLES

Nagios

- servers
- monitors
- Nagios
- amsterdam.obsidiansoft.com
- california.obsidiansoft.com
- obsidian
- check-host-alive
- Current Load
- Current Users
- Root Partition
- Swap Usage
- Total Processes
- SSH
- HTTP
- Mysql Obsidian connection

DEFINING DASHBOARDS

You can create as many dashboards as you like. Each is accessible by using its name as a parameter:

<http://.../obsidian/dashboard/?view=operational>

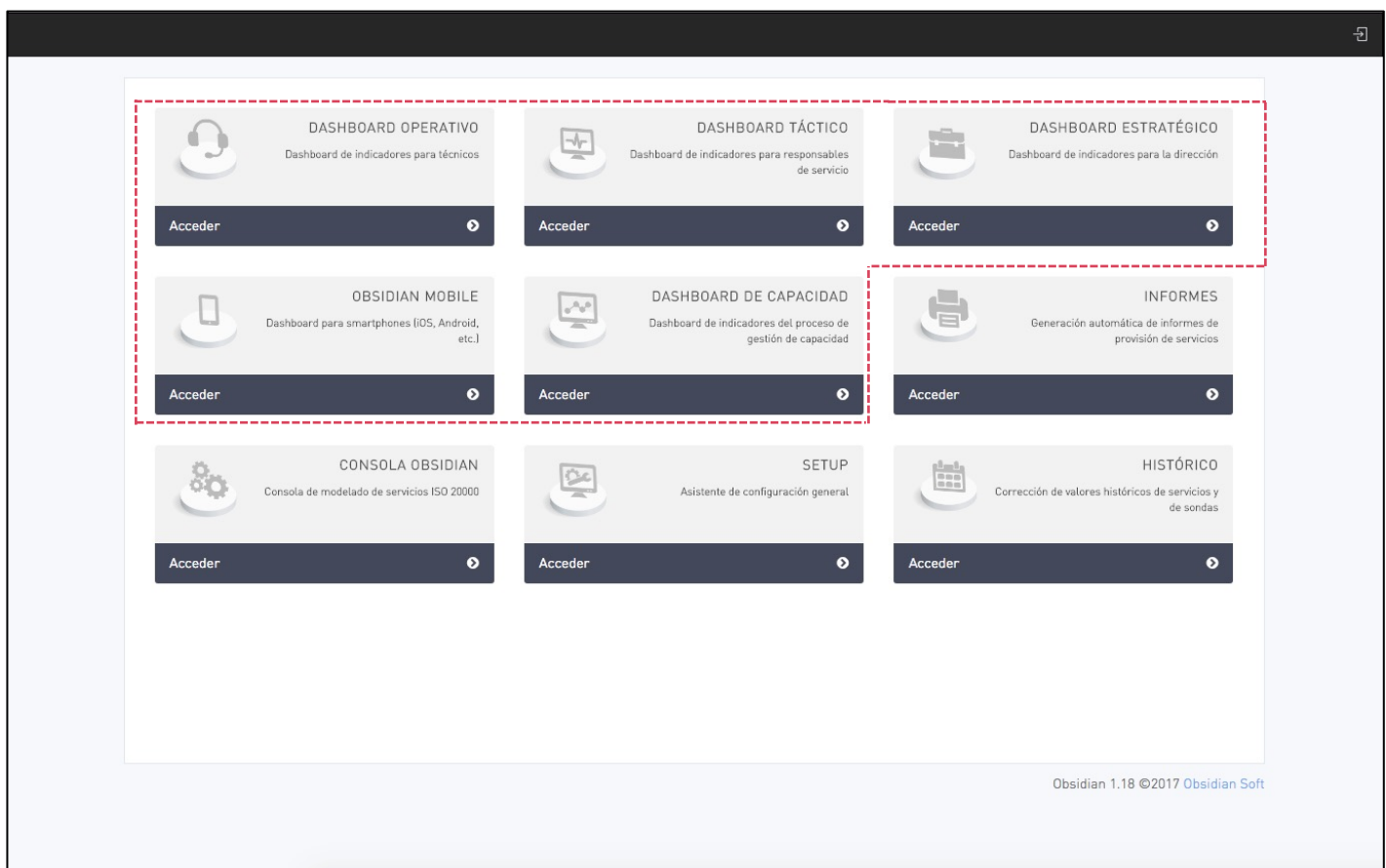
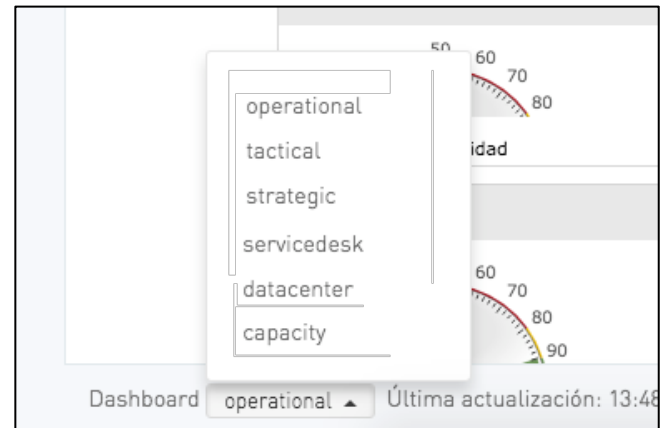
Default configuration includes the following dashboards:

- Operational dashboard
- Tactical dashboard
- Strategic dashboard
- Service specific dashboard example (Obsidian Monitoring service)
- Process specific dashboard example (Capacity)

Dashboards can also be accessed using shortcuts in the main menu

DASHBOARD SHORTCUTS

Dashboards are also accessible from the menu once advanced setup has been configured

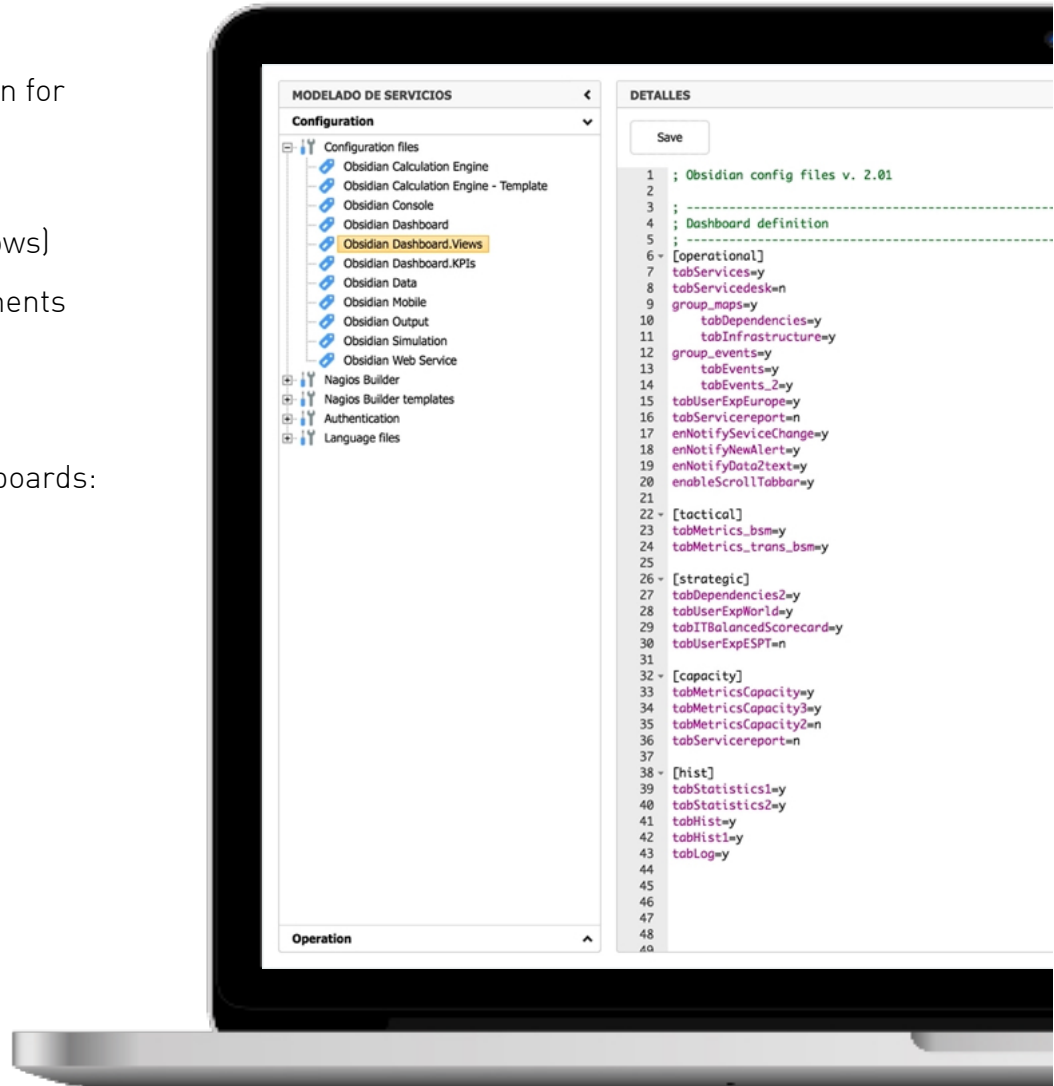


STRUCTURE OF DEFAULT DASHBOARDS

This section shows the default design for all main Obsidian dashboards:

- Dashboards are organised in tabs
- Tabs are organised in sections (rows)
- And sections are made up of elements

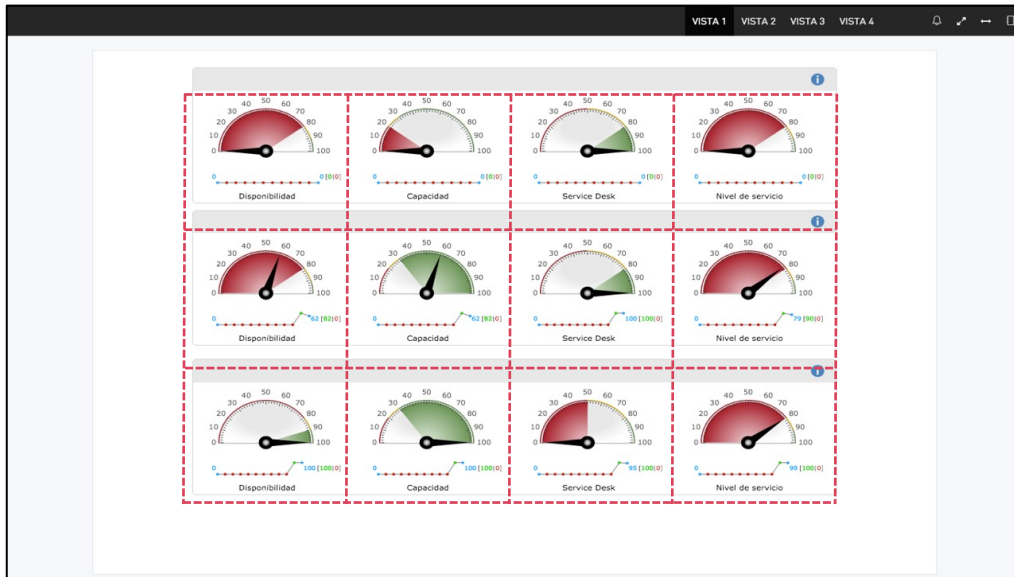
This is the definition of default dashboards:



TAB DEFINITION - ORGANIZATION OF ELEMENTS

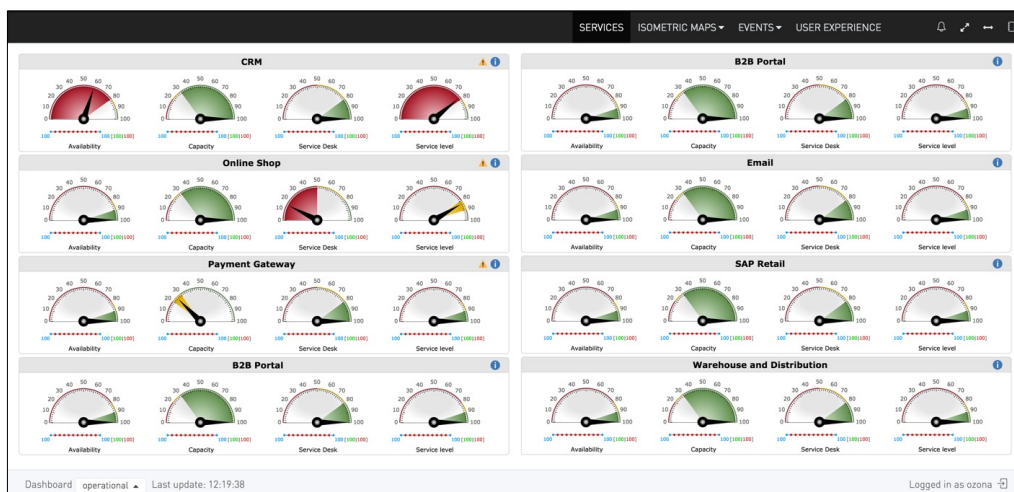
Each dashboard can have as many tabs as desired

Each tab is shown in a 4x3 grid space (without scroll) or 4xX (with scroll)



On Full HD or higher resolution screens, two 4xX columns will be shown.

If the number of services is more than 8, automatic rotation of services can be configured.

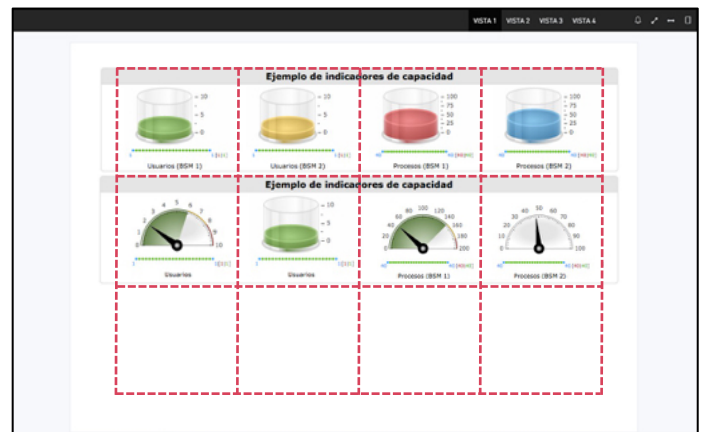


TAB DEFINITION - TYPES OF ELEMENTS

Despite the grid displaying 4 elements per row, it is also possible to show 1, 2 or 3 elements per row.



Many different types of elements can be used to represent the indicators.



The same metrics and indicators can be visualized in multiple ways. In the following figure, the data shown in all three rows is the same.

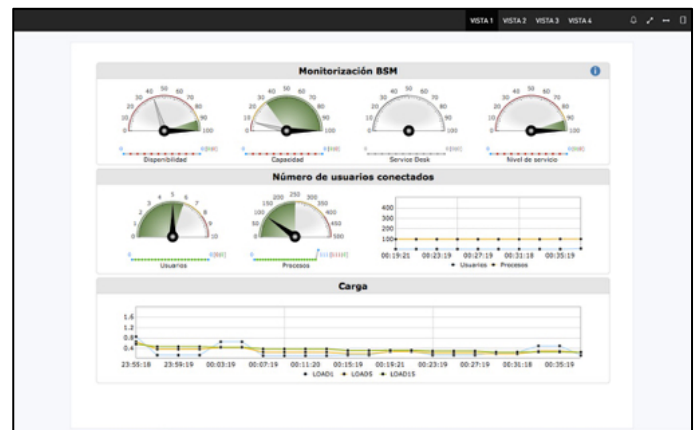


TAB DEFINITION - DEFAULT PATTERNS

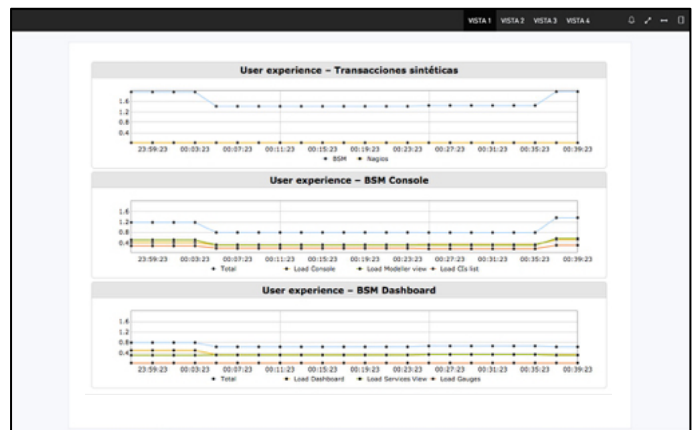
Default patterns usually will have 0, 1 or 2 metrics + progress chart (automatic update every X minutes)



Combination of a service indicator row + specific service metrics rows



Parsing one same metric, various values and the maximum reference value can be extracted



DIFFERENT WAYS OF DEFINING THE SAME METRICS

The three rows shown in the following figure represent the same data, with different visual presentations.

Indicator thresholds have not been defined.



The same data shown with custom thresholds



5

CONFIGURATION AND ADMINISTRATION



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

REFERENCE MANUAL

Obsidian's technical documentation includes a complete reference manual, how-to's for more common activities and REST API documentation.

REFERENCE MANUAL

Obsidian v2.0 Reference Manual

© 2016 Obsidian Soft

1. Dashboards Settings Reference

This page documents Obsidian configuration settings. They can be set in `/opt/ozona/obsidian/config/dashboard.views.ini`.

Editing configuration files

Configuration files, once edited, won't be replaced when updating the Obsidian RPMs to preserve customer's customizations. However, some new features may require the modification of the configuration files.

- Newest versions will be saved in `/opt/ozona/obsidian/config/` with the extension `.rpmnew` and they can be checked for reference
- Configuration files are available from the Obsidian Console selecting Configuration • Configuration Files • Obsidian Dashboard Views.
- After editing a configuration file, it can be saved by clicking:
 - the **Save** button over the editor
 - Ctrl + S** (Windows)
 - ⌘ + S** (Mac)

1.1. Views

A view is just a sequence of tab definitions defined in a `[section]` of the dashboard views config file. Views can be referenced as `http://obsidian_app/obsidian/dashboard/#view=section`.

Tabs may be enabled `TabCustomery` or disabled `tabCustomem`. Disabled tabs won't be visible.

⚠ Each tab has to be defined in the same config file. The parser will check if every referenced tab is properly defined.

Tabs may be grouped. Groups also have to be defined in the same config file. The parser will also check if every referenced group is properly defined. View definitions may also include comments. Comments start with `;` and may be a full line or part of a line.

```
[operational]
tabServices-y
tabServiceidm-n
; this is a group of two elements
group_napry
  tabdependencies-y
  tabInfrastructure-y
; this is another group
group_events-y
  tabEvents-y
  tabEvents_2of-y
  tabUserExpireupery
  tabServiceReport-n
  enotifyServiceChange-y
  enotifyNewAlert-y
  enotifyData2Text-y
```

ⓘ Groups are optional

Name	Description
<code>enableScrollTabbar</code> Deprecated	Parameter <code>enableScrollTabbar</code> and the even older <code>scrollTabbarenable</code> are deprecated and are no longer required. Menus get automatically stacked when required

REST API DOCUMENTATION

Table 12. `disableData2Text`

Description	Disable data2Text features for dependencies views that don't include information at the service level
Required	Optional
Valid values	<code>y</code> <code>n</code>
Default	<code>n</code>

Table 13. `enableStateOk`

Description	Highlight elements which are in an OK state
Required	Optional
Valid values	<code>y</code> <code>n</code>
Default	<code>n</code>

Table 14. `img`

Description	background image (PNG or SVG) for the dependencies map
Required	Mandatory
Valid values	<code>a valid image filename in /opt/obsidian/bsn/dashboard/images</code>
Default	<code>none</code>

Table 15. `imgHD`

Description	background image (PNG or SVG) for the dependencies map in FullHD displays
Required	Mandatory
Valid values	<code>a valid image filename in /opt/obsidian/bsn/dashboard/images</code>
Default	<code>none</code>

Table 16. `monitorType`

Description	Set what to show in the popup attached to an element in the dependencies view
Required	Mandatory
Valid values	<code>1 (availability)</code> <code>2 (capacity)</code> <code>3 (service desk)</code> <code>4 (worst of events)</code> <code>5 (worst of 1/2/3)</code> <code>6 (service level)</code>
Default	<code>4</code>

Table 17. `serviceID`

Description	ID of the dependencies map to be used
Required	Mandatory

6

INTEGRATION WITH DATA SOURCES



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

INTEGRATION MECHANISMS

Primarily there are two integration mechanisms:

- Active integration:
 - Via Obsidian's active access to databases or the tool's web services.
 - Suitable for tools that provide data for numerous monitors and raw data, such as monitoring tools.
- Passive integration:
 - By sending indicators to the Obsidian Web Service.
 - Suitable for service desk tools and other tools that occasionally send data or processed metrics.

NOTE

Documentation on integration with data sources is included as part of Obsidian's technical documentation and is not included in this user manual.

Obsidian v2.0 Integration Manual

© 2014 Obsidian Soft

There are two ways to integrate source tools with Obsidian:

- Inbound Integration
 - Active Inbound Integration, implementing a PHP class to allow the use of the source data with the rest of the Obsidian code
 - Passive Inbound Integration, sending data through the Obsidian web service
- Outbound integration
 - Direct access to the Obsidian database
 - Consuming the Obsidian web service

1. Active Inbound Integration

Obsidian can use information coming from any corporate data source, as long as the data is normalized to be used in conjunction with the rest of data sources.

For that purpose, Obsidian has two abstract PHP classes defined in `/opt/ozon/obsidian/includes/classes/`:

- `abstractIntegrator`: Abstract class that defines the interface to integrate data and monitoring information from any corporate source tool, such as monitoring tools.
- `abstractCMDBIntegrator`: Abstract class specifically designed to integrate CMDB tools that provide the structure of the services rather than monitors.

In order to integrate a new source tool with Obsidian, a new child of one of the abstract classes has to be implemented.



Some post-processing may be included in the parent classes to make sure that all the code is normalized.

1.1. AbstractIntegrator

Obsidian can use information coming from any corporate data source, as long as the data is normalized to be used in conjunction with the rest of data sources.

Name	Description
<code>__construct</code>	Constructor of the class. It should receive all the parameters needed to setup the integration
<code>getFirstLevel</code>	Provides the list of configuration items for whom the source tool provides information
<code>getSecondLevel</code>	Provides the list of configuration items for whom the source tool provides information
<code>getToolIcon</code>	
<code>getToolId</code>	
<code>getToolText</code>	
<code>getValue</code>	Function to retrieve the last valid value of a given monitor

`__construct`

Constructor of the class. No need to inherit parent class constructor

Name	<code>__construct</code>
Description	Constructor of the class. It should receive all the parameters needed to setup the integration
Required	yes

`getFirstLevel`

This method provides the list of configuration items for whom the source tool provides information

Name	<code>getFirstLevel</code>
Description	Provides the list of configuration items for whom the source tool provides information
Required	yes

`getSecondLevel`

This method provides the list of metrics and indicators available for a given configuration item

The second level may contain several inner levels. It is up to the integration class to organize it in just one level or several

Name	<code>getSecondLevel</code>
Description	Provides the list of configuration items for whom the source tool provides information
Required	yes
Parameters	id of a configuration item in the format <code>server_name_or_ip+monitor_id</code>



7

SIMULATION SCENARIOS

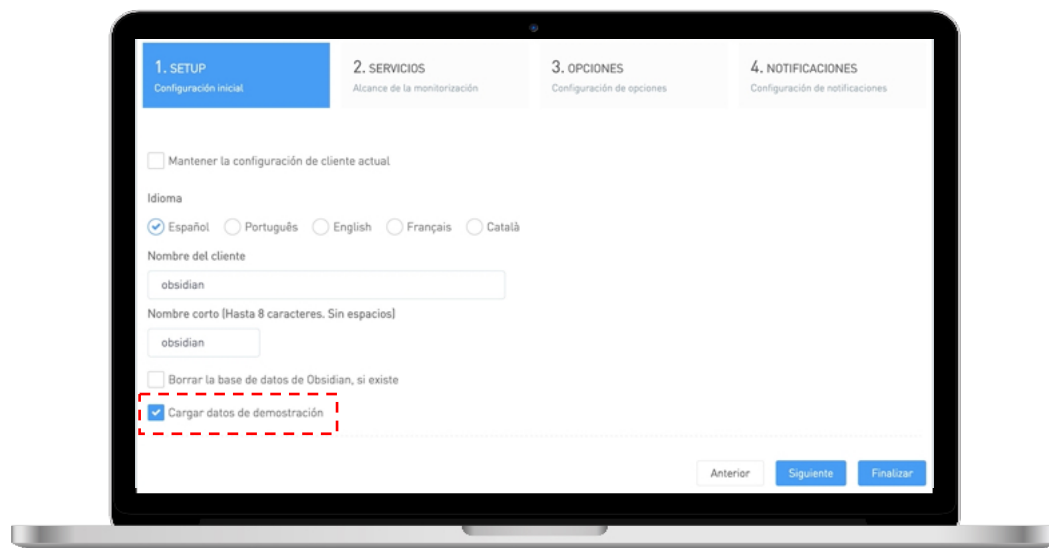


OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

CONFIGURATION OF SIMULATION SCENARIOS

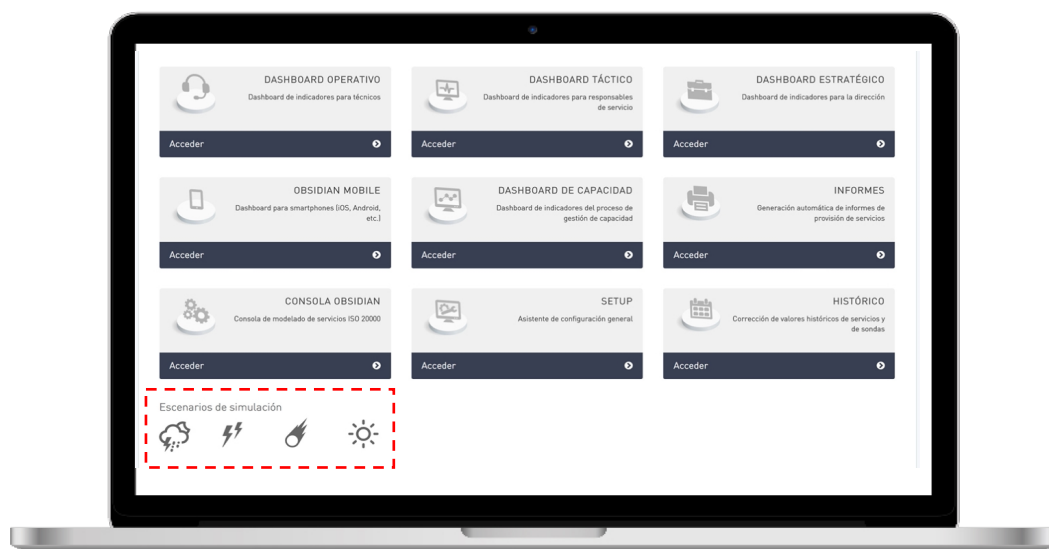
Simulation scenarios can be activated from the Obsidian appliance main menu, once demo data has been loaded.

In addition to default simulation scenarios, which are used for demos and training, it's also possible to define your own simulation scenarios, simulating any unavailability or service level alert, CI or monitor.



There are 3 default scenarios which you will see in the following pages.

The last scenario marked with a sun shows normal performance without any active simulation scenario.



SCENARIO 1 - MULTIPLE ALERTS


There are multiple alerts in the service infrastructure. However, they don't affect the services, which remain at an adequate service level.

OBJECTIVES

Show how availability management based on infrastructure alerts does not always show a realistic overview of services.

SCENARIO ACTIVATION

Scenarios are activated from the main menu:

1. Load the  simulation scenario in the appliance main menu.
2. Access the Strategic dashboard (Business view)
3. Access the Operational dashboard (Technical view)

KEY POINTS

- There are various services with alerts
- However, none of them are causing impact at service level

SCENARIO 2 - FIREWALL DISABLED


The Dublin datacentre firewalls have been reconfigured and an error has occurred meaning that traffic is being redirected, impeding the correct external routing.

OBJECTIVES

Show how an element's unavailability causes the unavailability of a service and how that affects many dependant services.

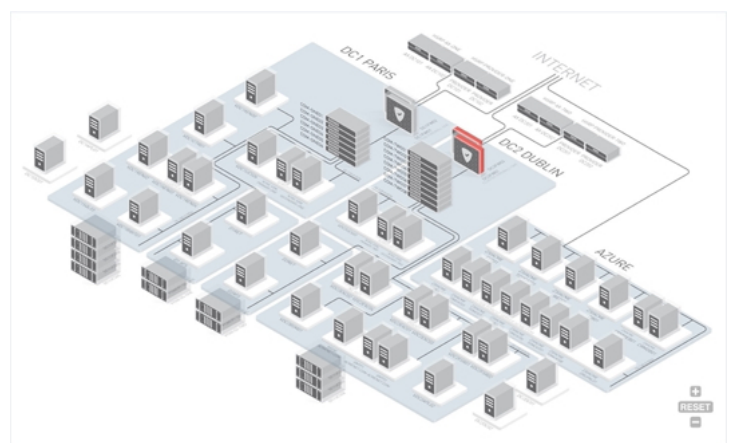
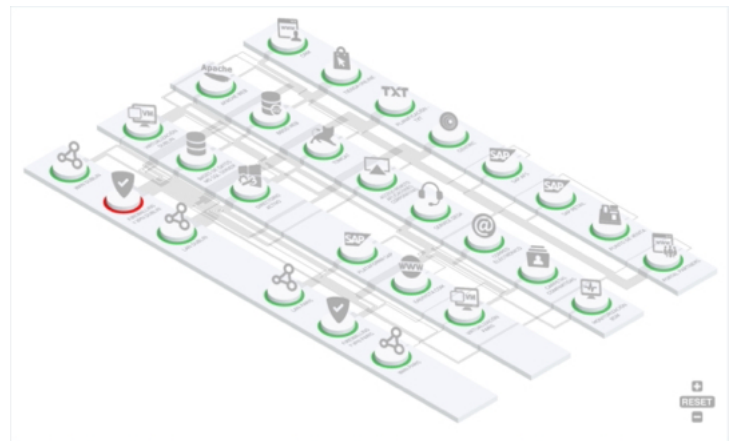
SCENARIO ACTIVATION

The scenario is activated from the main menu:

1. Load the  simulation scenario in the appliance main menu
2. Access the Strategic dashboard (Business view)
3. Access the Operational dashboard (Technical view)

KEY POINTS

- There are multiple impacted services
- However, only one service is affected, which is the "firewalling and Dublin VPN"
- And from this service, only the firewall is down



SCENARIO 3 - PRO-ACTIVE CAPACITY MANAGEMENT

The Dublin datacentre firewalls have been reconfigured and an error has occurred meaning that traffic is being redirected impeding the correct external routing.

OBJECTIVES

Show how an element's unavailability causes the unavailability of a service and how that affects many dependant services.

SCENARIO ACTIVATION

The scenario is activated from the main menu:

1. Load the ⚡ simulation scenario in the appliance main menu
2. Access the Strategic dashboard (Business view)
3. Access the Operational dashboard (Technical view)

KEY POINTS

- The webs have a minor capacity problem
- At a technical level, the only affected service is the "Tomcat" service
- And from this service, various machines from the "Tomcat" cluster have performance problems.



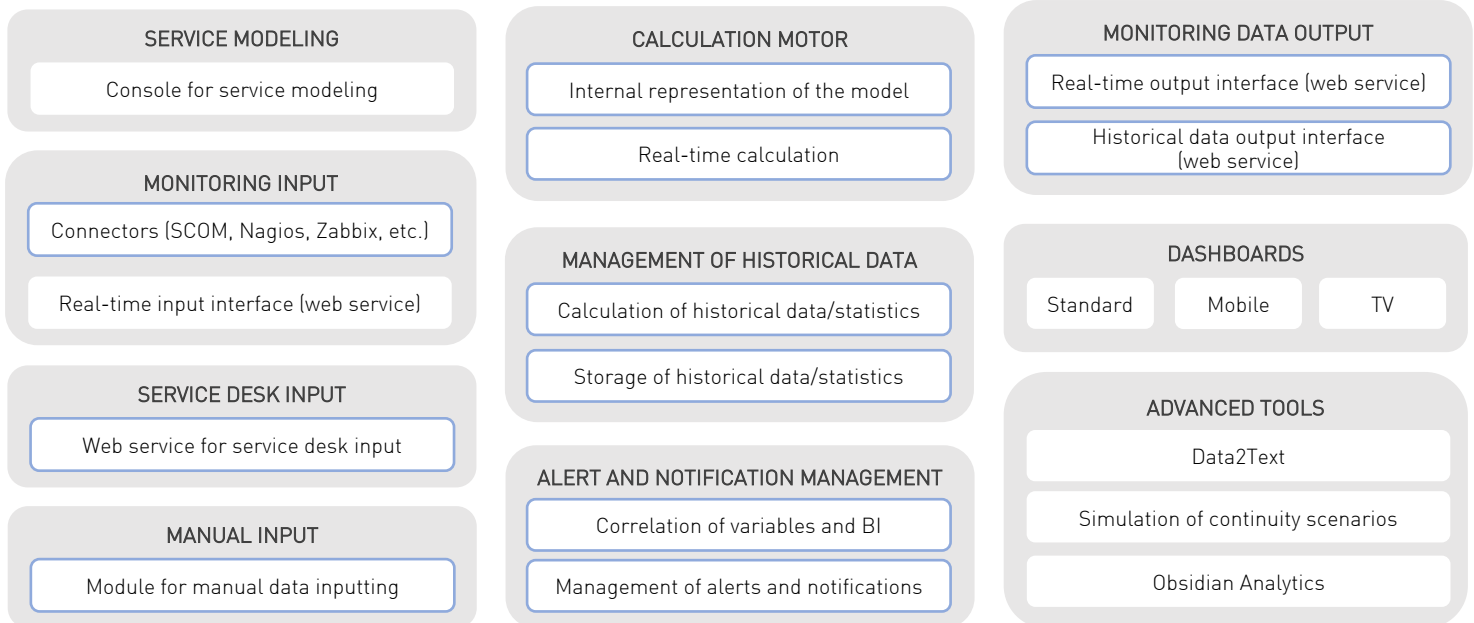
8

**OBSIDIAN
ARCHITECTURE**



OBSIDIAN
BUSINESS SERVICE INTELLIGENCE

OBSIDIAN ARCHITECTURE




A network diagram background consisting of a complex web of interconnected nodes and lines. The nodes are represented by small circles in various shades of blue and white, connected by thin, light blue lines. The diagram is more dense and detailed on the left side of the page, fading towards the right.


OBSIDIAN


BUSINESS SERVICE INTELLIGENCE

www.obsidiansoft.com

Madrid
C/General Martínez Campos, 44 · 28010 · Madrid
Tel: +34 91 781 09 57

 Obsidian Soft

 obsidiansoftware

 @ Obsidian_Soft