software AG
GOVERNMENT SOLUTIONS

IoT platform

Cumulocity IoT - product description

Fact sheet

1. Cumulocity IoT – Components and Capabilities

Cumulocity – Device Integration Platform

Cumulocity is an independent device and application management Internet of Things (IoT) platform. It connects and manages devices and assets efficiently, controlling them remotely.

- Connect devices and assets over any network
- Monitor conditions and generate real-time analytics
- React immediately to conditions or situations

Overview

Cumulocity gives very fast visibility and control over remote assets, be these houses, cars, machines or any other assets that needing to be managed.

Cumulocity provides:

- Certified hardware kits and software libraries that can be used to bring remote assets into the cloud
- Device management, data visualization and remote control functionality through the web
- Rapid customization of the above through Cumulocity Event Language rules and Cumulocity applications
- APIs for extending the existing functionality or interfacing Cumulocity with other IT services such as ERP or CRM systems. Cumulocity can also host HTML5 application.

All this is provided through a cloud-based (or on-premise) subscription service making the creation of Internet of Things (IoT) solutions with Cumulocity fundamentally different from bespoke development and RAD (rapid application development). Users can start immediately with a large amount of existing functionality without worrying about IT infrastructure (hosting, networking, security, storage and backup) and IT management (all software is available to users).

Cumulocity works with any network architecture, but is specifically designed to work out of the box with mobile networks. In the following sections, we will give a short overview of the different functional areas with references to more detailed descriptions.

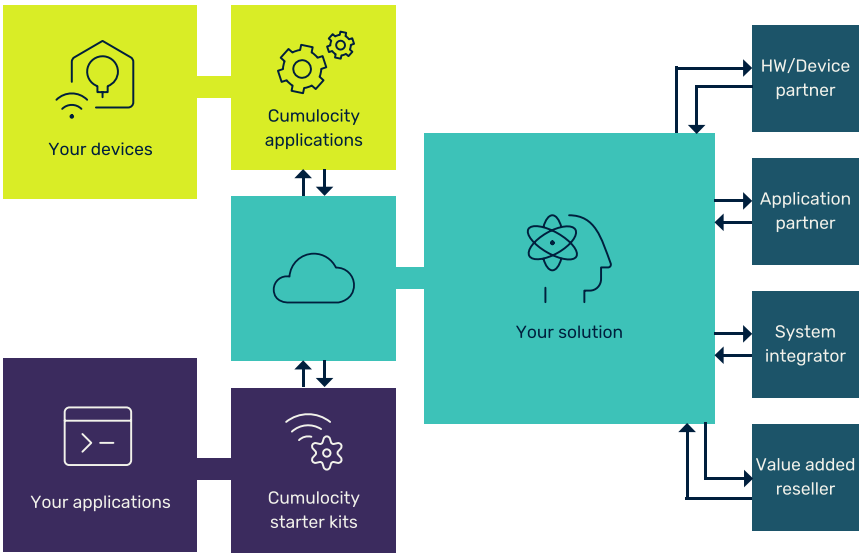


Figure 1: Hardware kits and software libraries

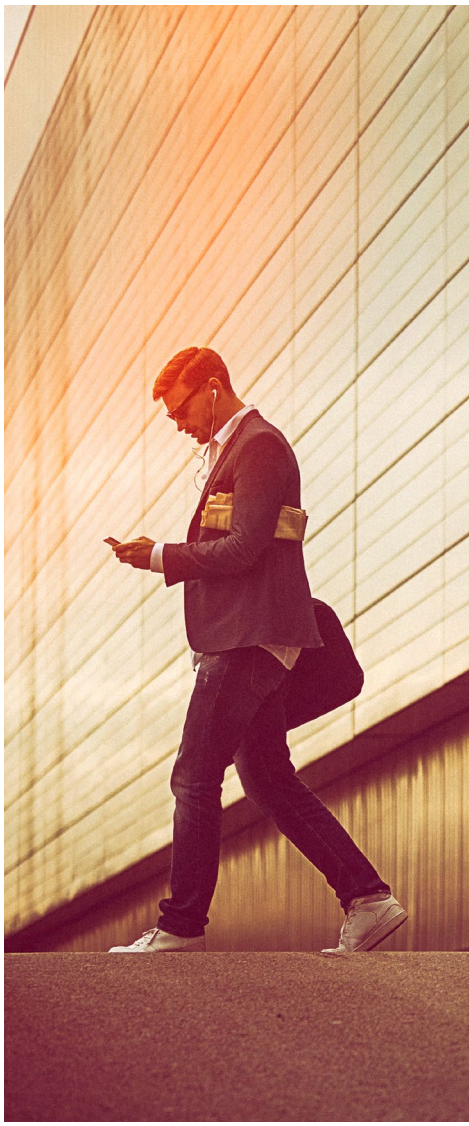
Functionality within Cumulocity

Cumulocity directly supports various devices with ready-made software libraries and examples. These can be specialized devices for a particular use case, as locations trackers, OBUs and vending telemetry devices. They can also be developer kits for building generic devices, such as Arduino, Raspberry Pi, Cinterion boards, Tinkerforge sensors and more. These developer kits are described in more detail in the corresponding chapters of the “Devices” section in this documentation.

Outside of the specific kits, many other devices can run the software with no or little modification. That is why the software is provided in source code form to extend it to any other device. There are also generic client libraries for Java, Java ME, C/C++ and Lua for own implementation. If a device uses a completely proprietary runtime environment, Cumulocity’s REST resp. HTTP interfaces can always be used. These will work on practically any Internet-connected device today, down to the smallest systems.

Mobile networking support
Cumulocity supports any type of Internet connectivity in a secure manner. It gracefully deals with intermittent, bandwidth-restricted and unidirectional connections (such as communication through NAT). Where desired, Cumulocity can control remote devices in a real-time manner.

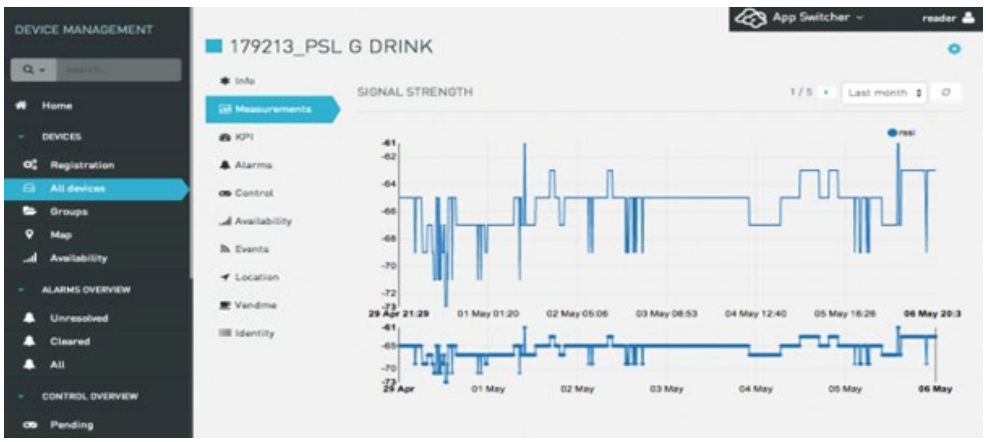
Mobile Internet connectivity is an ideal choice for many machine-to-machine applications, since it works well nearly everywhere without requiring any integration with a company's network infrastructure. This is especially true if an M2M SIM card allows for free roaming between mobile networks. The large bandwidth requirements of consumer applications are often not required. With Cumulocity, you can benefit from mobile connectivity without requiring additional network provider services such as VPNs and public or even static IP addressing.



Device management
Cumulocity provides extensive device management for fully certified devices.
This includes hardware and modem information:

- Connection monitoring
- Centralized fault management and service level monitoring
- Configuration management
- Software and firmware management
- Graphs of device statistics
- Frequently used remote controls (e.g., restart button, switches)
- Troubleshooting features such as event list and operations queue

The level of depth in device management may depend on device features. (e.g., if a device does not support remote firmware upgrade, it will also not be available through Cumulocity.) For interfacing devices not yet certified with Cumulocity, the Device Management Library and the REST Developer's Guide is publicly available.

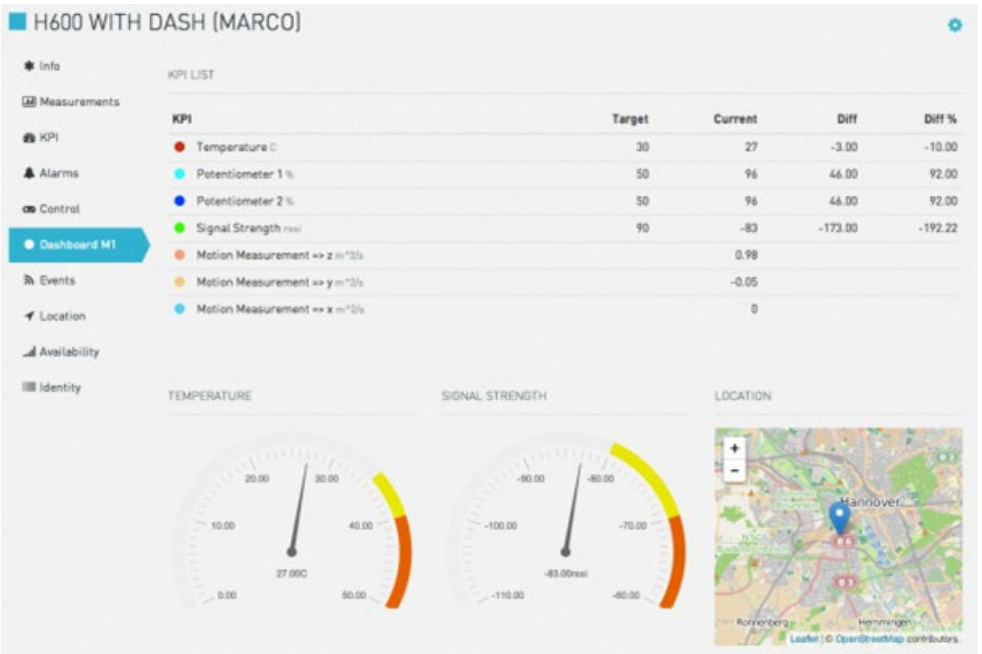


Visualization and remote control
Cumulocity visualizes sensor data centrally and graphically through its modern web user interface. It also exposes common remote controls to users with the relevant permissions.

The user interface automatically adapts itself to the connected devices - no configuration required. For example, a connected device that supports being restarted from remote shows a “Restart” button. If the device sends light sensor data, it shows a graph with the readings from the sensor.

It also adapts itself to the web browser that is being used. For example, a mobile phone or tablet with limited screen size, it will change user interface controls to use less screen estate.

Through the Sensor Library, common sensor and control types are rendered correctly regardless of the device that produces the sensor data.



Customization
The functionality described above already provides a wide range of device management, visualization and control options. Furthermore it produces custom visualization, new control widgets and custom business logic. Cumulocity has extensive customization options:

- Write alarm rules to reprioritize or suppress alarms and to define SLA parameters
- Use Cumulocity Event Language to implement real-time business rules. For example, get an email when critical events happen, or trigger automated actions on devices in that case
- Set up a graphical dashboard with most important KPIs
- Subscribe to plugins that contribute new functionality to the Cumulocity application

APIs
Cumulocity exposes its complete functionality through programming interfaces (APIs).

This means that all of Cumulocity's functionality is available for using in different contexts outside of what Cumulocity directly provides—in own applications, in own devices.

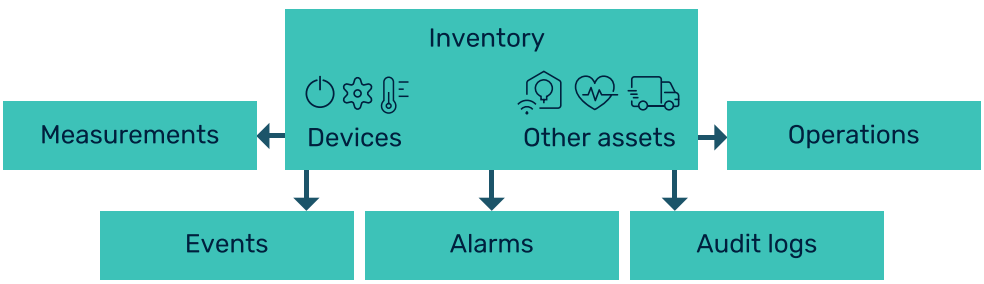
In contrast to many other M2M and IoT platforms, Cumulocity uses the same APIs and the same interface technology for all use cases. As a consequence, that provides a wider range of choices in putting intelligence into IoT devices, depending on how powerful they are. Only one set of APIs and one technology need to be used to build a complete solution from device to application.

Cumulocity uses HTTP and REST, which is today the most widely used interfacing technology working on any Internet-connected device ranging from small embedded microcontrollers up to desktop PCs. The secure variant, HTTPS, is used for the most security critical applications and will give the best possible security.

The plugin concept of Cumulocity enables to write new user interface functionality that will seamlessly extend the existing Cumulocity application.

Cumulocity's domain model

Cumulocity captures all relevant aspects of devices and assets in the Internet of Things.



- The inventory stores all master data related to devices, their configuration and connections. It also contains all related assets (like vehicles, machines, buildings) and their structure. Measurements contain numerical data produced by sensors (like temperature readings) or calculated data based on information from devices (service availability of a device).

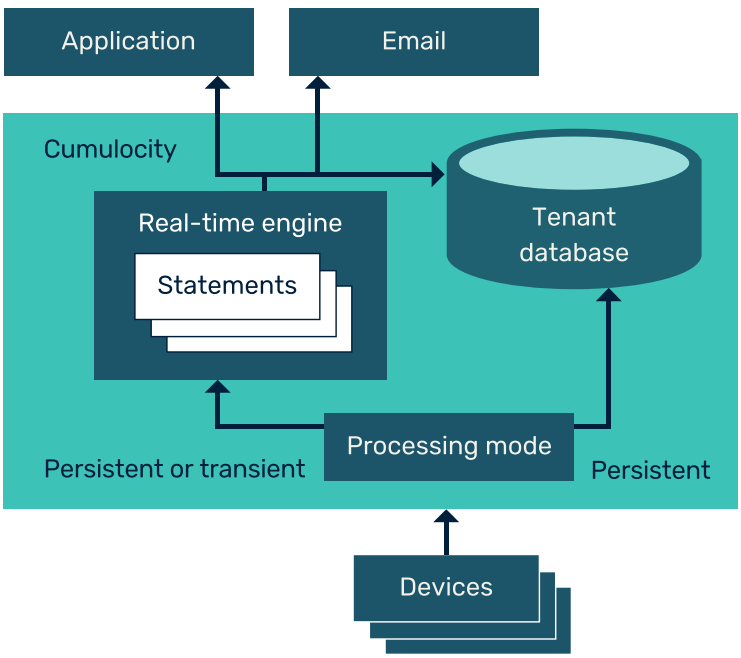
- Events contain other real-time information from the sensor network, such as the triggering of a door sensor. Events can also be alarms. The user or operator of the system has to take action to resolve the alarm (like a power outage). In addition, security-related events are shown as audit logs.
- Operations relate to data that is sent to devices for execution or processing, such as switching a relay in a power meter or sending a credit to a vending machine.
- One of the great innovations in Cumulocity is its standardized representation of common devices and sensors as well as concepts for flexibly extending and modifying this representation. By default, Cumulocity comes with detailed visualizations of sensors, smart meters, trackers and other devices. It has many options to fit in local customizations.

As a result, Internet of Things applications can be written independently from connected devices and underlying sensor networks, customized for specific cases in different web configurations or different devices from manufacturers.

Real-time processing

Cumulocity allows developers and power users to run real-time IoT business processes. The user can choose if data is stored on a permanent basis or is temporarily used to generate reports or analytics and then is deleted automatically. The processes and results update continuously.

Cumulocity has a real-time engine receiving all data coming from devices or other data sources for immediate processing user-defined business operations. These user-defined operations can alert applications of new incoming data, create new operations based on the received data (such as sending an alarm when a threshold for a sensor is exceeded), trigger operations on devices or send email. This operation logic is implemented in Cumulocity Event Language, a high-level domain-specific language designed for IoT real-time data.

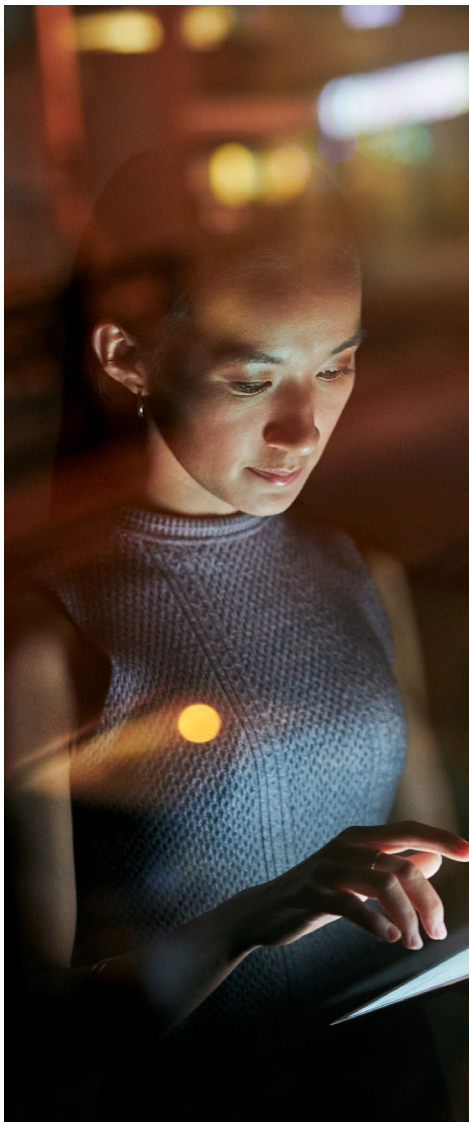


Cumulocity Event Language covers statements as illustrated in the following examples. They are grouped into units of deployment called modules. Modules can be deployed as part of a Cumulocity application. They can be edited with the Cumulocity administration application. Through the REST API, application developers can develop user-friendly domain-specific business operation wizards for their specific use cases. For example, a home automation developer can create a wizard providing thresholds for temperature sensors in order to control heating devices.

The image above also illustrates another feature of Cumulocity: the possibility to send data exclusively for real-time processing. Data marked as “temporary” is not stored into Cumulocity’s database but just handled by the real-time engine. This saves on storage and processing cost for example when tracking devices in real-time without requiring data to be stored.

Benefits of using real-time processing
Cumulocity’s real-time processing feature has the following benefits:

- React instantly to events from remote sensors
- Develop highly interactive IoT applications
- Run IoT use cases directly inside Cumulocity without software development and leave the hosting and management to Cumulocity
- Validate, normalize and derive data according to own business rules across different device makes
- Trigger automated remote control actions based on events.
- Use powerful, stream-oriented business logic, like time windows and joins.
- Reduce the cost of online tracking devices by preselecting data necessary for long-term storage

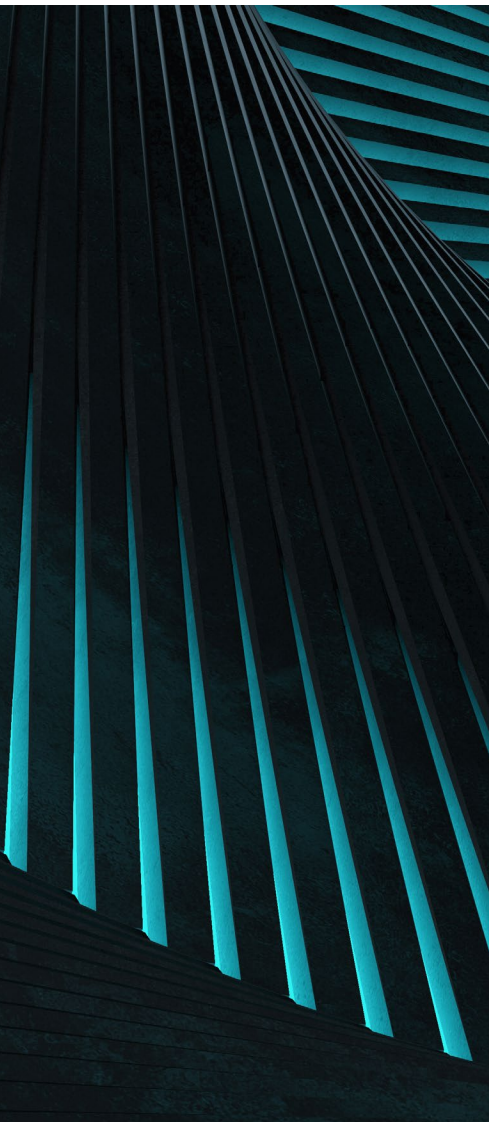


Developing applications

Cumulocity is designed to accommodate arbitrary vertical IoT applications in addition to its generic functionality. Applications are registered in Cumulocity either as “own” applications or “market” applications. An application can be any combination of a complete, standalone user interface application, a set of user interface plugins, or a set of statements in Cumulocity Event Language. With Cumulocity users can publish any software to other users or customers.

Tenants can subscribe to applications to get

- Extensions to the Cumulocity user interface
- Entirely new user interfaces
- Branding of the Cumulocity user interface
- New server-side business logic



Applications and subscriptions

Applications are registered in Cumulocity either as “own” applications or “market” applications. “Own” applications are only available to users of a particular tenant and are registered by the tenant’s administrator. Own applications are used, for example, during application development when you do not yet want to make a particular application version available for a wide audience. They are also used for functionality that is proprietary for an enterprise, for example, interactions with in-house IT systems.

“Market” applications are available to all tenants of Cumulocity. Subscribing a tenant to a market application makes this application available to the tenant. Applications are identified by a so-called application key. The application key enables Cumulocity to associate a request with one particular application.

An application can be any combination of:

- A complete, standalone user interface application regardless if based on the Cumulocity UI framework (see below) or any other web components
- A set of user interface plugins
- A set of statements in Cumulocity Event Language

User interface applications appear in the application switcher widget on the top right of Cumulocity, so that users can navigate between the subscribed applications. They can be hosted on an external website, in which case the application switcher just directs the user to that website. They can also be hosted through Cumulocity, in which case the application will be made available through a URL .cumulocity.com/apps/.

Security aspects

Cumulocity addresses security on various levels. All business partners and hosting services have recognized security certificates. Cumulocity also deals with network security aspects by individual authentication and authorization methods. Connections from and to Cumulocity are established using HTTPS technology. All tenants have full rights to add or terminate users and user groups. The tenant also assigns rights to agents and devices.

Cumulocity complies with Nokia Networks’ “Design for Security” policy (which is unfortunately not available publicly) and Deutsche Telekom’s “Privacy and Security Assessment” (PSA, detailed criteria in English detailed criteria in German).

Physical security aspects
Physical security of IT systems prevents unauthorized physical access to servers, storage, and network devices.

Cumulocity Standard Edition accounts are hosted at Amazon Web Services (AWS). AWS has been certified according to ISO 27001, DSS and other standards. It features extensive physical security measures and is independently audited. Not all details are published for actual security reasons. Audit reports can be obtained directly at AWS Compliance.
Our strategic hosting partners follow up-to-date concepts and concepts of data security.

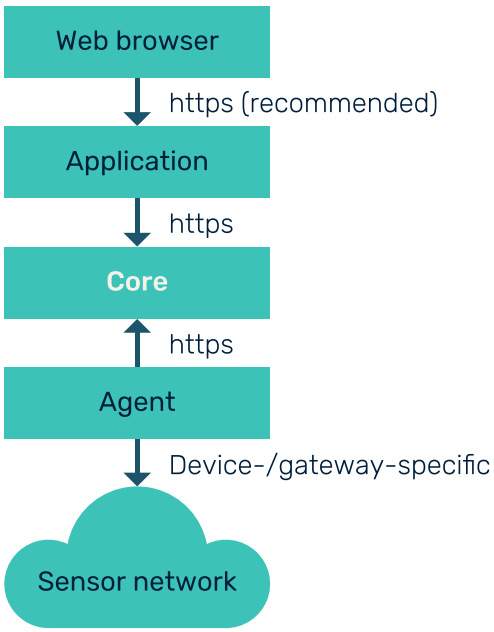
In IoT solutions, physical security also includes unauthorized access to IoT devices, for example, to redirect or manipulate data from devices, read credentials from devices or change a device’s configuration. We recommend reviewing the physical security of the devices that is planned to use for the IoT solution and, e.g., make configuration ports unavailable to unauthorized people or include tamper sensors as an additional security control within your own system. As the operator of the platform Cumulocity we do not control internal systems of our tenants. As a tenant you must follow a powerful and thought-through security concept for your own system.

Network security aspects

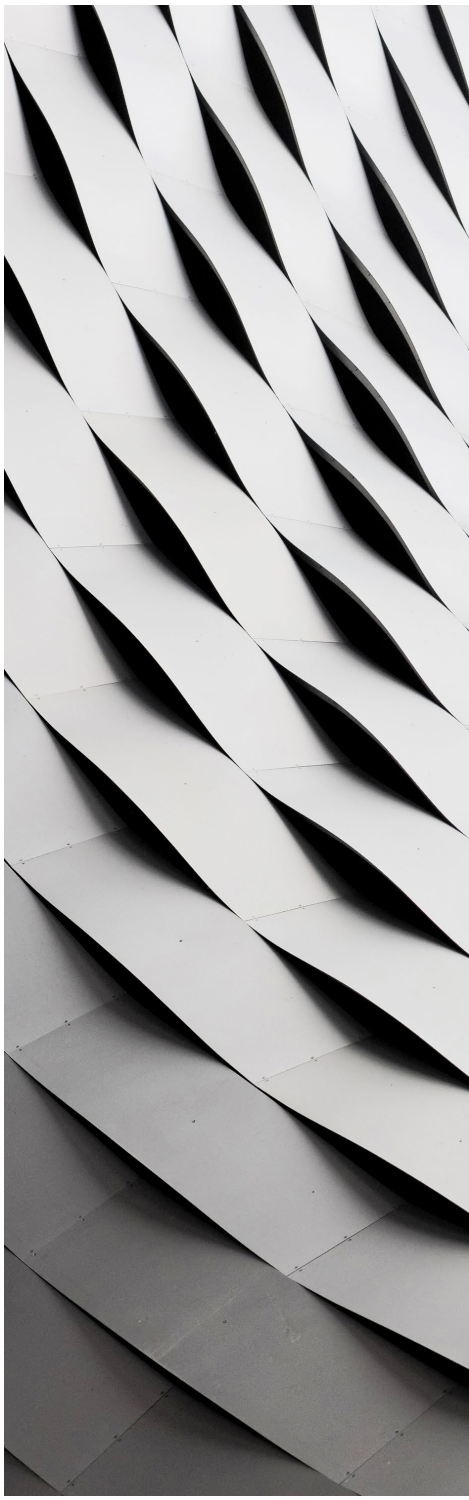
Network security prevents unauthorized access to data transmitted over the network and tampering with or unauthorized modification of data. It also ensures that network services are available.

Cumulocity ensures that your data stays confidential and cannot be tampered with through an end-to-end implementation of HTTPS from devices to applications. It uses up-to-date encryption technology that has been independently rated "A" by SSLlabs. Any communication with Cumulocity is subject to individual authentication and authorization.

This communication architecture is illustrated below. Inside the sensor networks and from the sensor networks to agents, device- and gateway-specific protocols may be in use (such as ZigBee or Modbus). Securing these is a device-specific matter. Agents communicate with the Cumulocity platform using HTTPS to send and receive data. Similarly, IoT applications use HTTPS for communication. If an IoT application exposes own interfaces towards web browsers, it is recommended that these use HTTPS. This way, the whole path from agents to the end user is secured.



As mentioned above, Cumulocity does not require any device that might expose ports or services on the Internet. This is an important feature: it not only simplifies the connection of devices to Cumulocity, but also simplifies the safety backup of these devices drastically. When deploying an IoT solution, check other services that might make a device available on the Internet or expose it, such as Web-based device managers or SMS-based configuration options.



Application security aspects

Application security addresses security at the software level. Cumulocity follows standard practices for application-level hardening as making sure that only properly upgraded operating systems and web servers are in use. A number of additional “best practices” are employed to make Cumulocity secure by design.

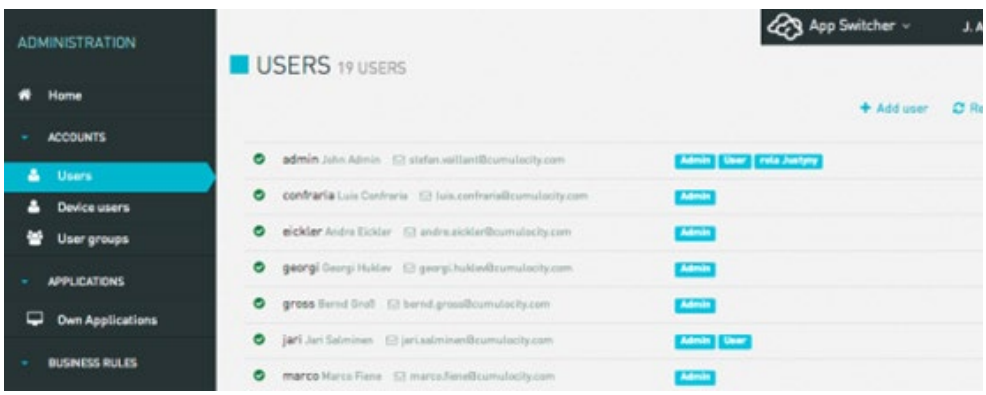
- All functionality of Cumulocity is coherently implemented with the same set of publicly documented, sessionless REST APIs. This means that none of the popular “session stealing” techniques will work with Cumulocity.
- Cumulocity does not use a SQL database for IoT data storage and is itself not based on a scripting language. This means that so-called “injection attacks” will not work with Cumulocity Applications and subscriptions.
- As discussed above, devices are clients at Cumulocity and therefore popular attacks to devices will not work. “Market” applications are available to all tenants of Cumulocity. Subscribing a tenant to a market application makes this application available to the tenant. Applications are identified by a so-called application key. The application key enables Cumulocity to associate a request with one particular application.
- Devices are individually connected with Cumulocity’s device registration feature. This means that if a device is stolen or tampered with, it can be individually disconnected from Cumulocity.

Access control

Cumulocity uses a standard authentication and authorization process based on realms, users, user groups, and authorities. A realm is a database of users and user groups, who follow the same authentication and authorization policy. A user is a person or an external system entitled to access protected resources inside Cumulocity. Access is controlled through permissions. For simplifying administration, users can be grouped into user groups sharing similar permissions. A user can be a member of several user groups so that the user has the combined permissions of the groups.

Cumulocity creates a new realm for each tenant to store the users of that tenant. Realms provide an own namespace for usernames, allowing users to keep the names that they are familiar with from their own enterprise IT or other IT systems. There is no conflict between user names: a user “smith” of one particular tenant is different from a user “smith” of another tenant. This username is valid for all Cumulocity applications that a tenant subscribes to.

An initial administrator user who can create further users, user groups, and can assign permissions to these users and user groups, automatically populates each new realm. This enables an enterprise to manage users and their permissions on their own using the administration application.



The ability to execute certain functionality on the system depends on two concepts: permissions and ownership. Permissions define explicitly what functionality can be executed by a user. Cumulocity distinguishes read permissions and administration permissions. Read permissions enable users to read data. Administration permissions enable users to create, update and delete data. Read and administration permissions are separately available for the different types of data in Cumulocity. For example, there are read permissions for inventory data, measurements, operations and so forth.

Objects in the inventory also have an owner associated with them. Owners can always, regardless of their other permissions:

- Read, update and delete the inventory objects they own
- Create, read, update and delete data associated with the objects they own

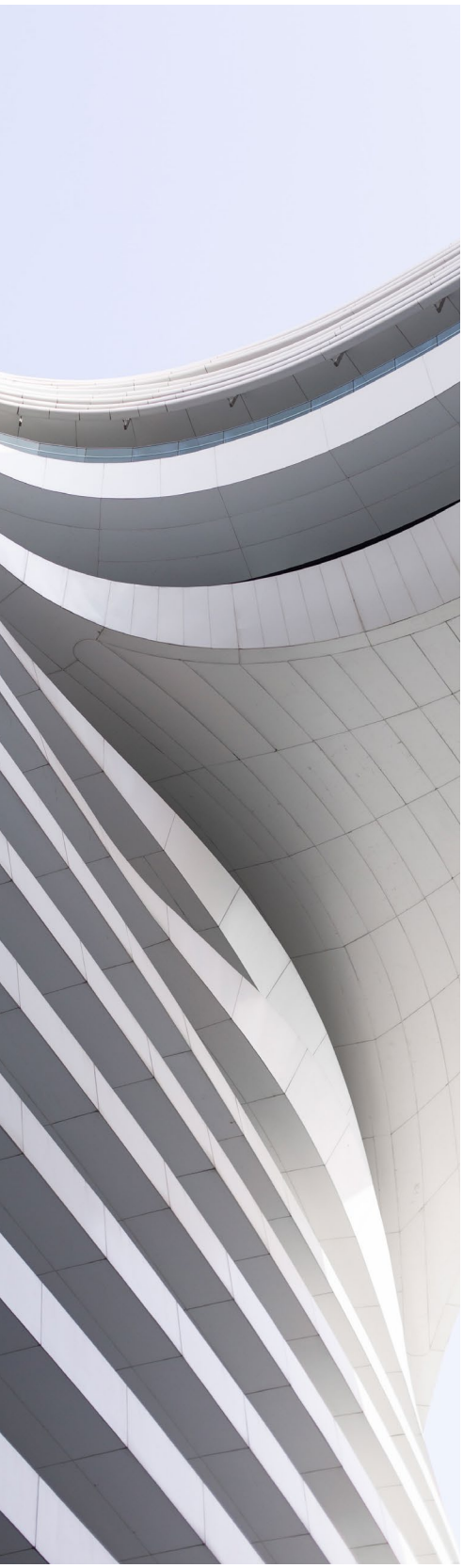
For example, if you are the owner of a smart meter in the inventory, you can store meter readings for that smart meter even if you do not have any other measurement permissions.

The inventory also features a "create" permission. A user having just the "create" permission can store new objects in the inventory, but cannot read, modify or delete any other data. This is mainly relevant for devices. The "create" permission also includes the possibility to link an object to another object as child device or child asset.

Apama

The Apama platform allows you to analyze and act on high-volume business operations, device and customer interactions in real time—and identify what is likely to happen and still influence the outcome. Software AG's Apama support streaming and predictive analytics, combining event processing, messaging, in-memory data management and visualization. This platform is the most complete solution to turn relentless data streams—like those produced by the Internet of Things (IoT)—into meaningful real-time metrics.

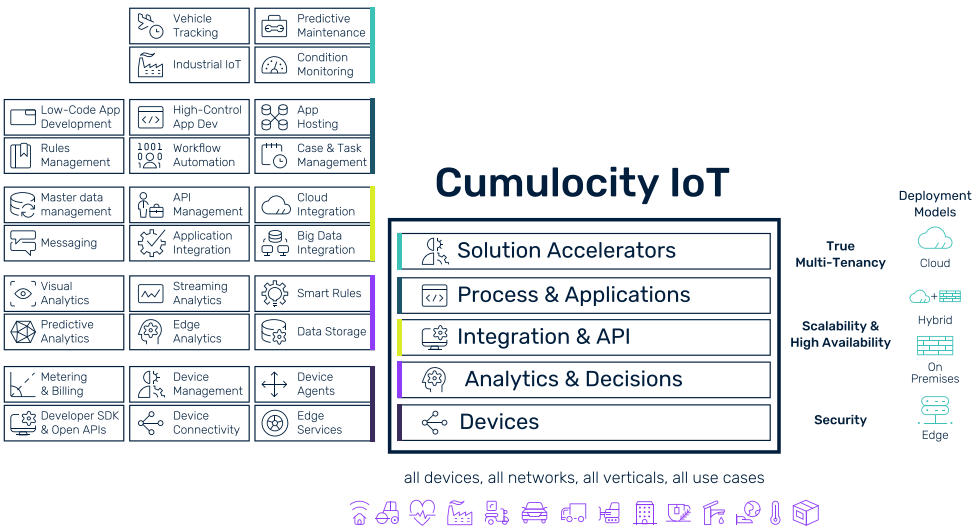
- **Streaming analytics:** Apama can watch, in real time, for evidence of the patterns and trends that your data mining tools have identified. The instant a streaming analytical tool encounters a data point or an event associated with one of the patterns or trends you are watching for, it can raise an alert or trigger an action. For example, after performing a deep analysis of engine maintenance records, an automobile manufacturer might discover a pattern showing a higher incidence of costly engine repairs in cars where the engine oil has degraded past a certain viscosity level. Instead of recommending that owners change the oil every 5,000 miles, the manufacturer decides to recommend owners change the oil when it reaches a specific viscosity level (as identified by the analysis of the historical record). To facilitate this new recommendation, the manufacturer decides to implement a streaming analytical system that can monitor sensor data pouring in from each car's onboard computer. The moment the streaming analytical system detects that a car's oil has reached that critical viscosity point—regardless of miles driven—it will send a message to the dealer responsible for servicing that car and prompt the service manager to contact the owner to schedule an oil change.
- **Predictive analytics:** it can help an organization seize opportunities that might not be apparent in a traditional analysis of the historical record—in part because predictive analytical tools are not limited to the structured data which most traditional analytical tools focus on. Predictive analytics can predict outcomes that take into consideration unstructured data as well, including images, video, social post, and audio data. And, because of the non-deterministic approach inherent in predictive analytics, predictions can be made based on patterns and events that the tool discovers on its own in the course of its analysis. Indeed, the patterns that predictive analytics can find and act upon may be far more subtle and complex than those described by traditional data mining approaches.



While the combination of historical and streaming analytics can deliver significant benefits, several important use cases remain in which key opportunities or risks will be insufficiently identified. Consider again the example of a car's motor oil. While streaming analytical systems can issue an alert as soon as they detect that a car's oil has reached a specific viscosity level, streaming analytical systems are not designed to predict when a given car's oil viscosity will reach that point. That lack of advanced insight could be problematic: if a car owner drives 200+ miles every day, the viscosity of that car's oil will change rapidly, and though the streaming analytical system can detect that the car should have its oil changed, it will not alert the local service manager to schedule the car for service sooner rather than later. If the service manager schedules the oil change 30 days in the future, the car's oil may have degraded completely, and the engine may be irreparably damaged before it arrives for scheduled service.

Key Capabilities

The Software AG's digital business platform provides key capabilities in context of a digital platform for artificial intelligence and technology development.



IoT Framework (IaaS, PaaS, SaaS)

IaaS Level

Cumulocity platform can be based on **AWS** (Amazon Web Services) as Infrastructure-as-a-Service (IaaS) and use AWS's native VPN technology (e.g., «VPN Gateway») to provide VPN connectivity between the cloud computing domain and local (on-premise) systems.

PaaS/SaaS & Presentation Level - IoT Capabilities

«Cumulocity IoT Core» will be the central component providing from basic to very advanced IoT functionalities. Basic capabilities include:

- Device connectivity
- Device management
- Scalable storage

- Real-time and streaming analytics provided by the **Apama** component.
  For performance reasons, these functions are realized in a separately scalable component with bidirectional interfaces to the Cumulocity IoT core. Apama executes decision and AI/ML logic based on:
    - o EPL (Event Programming Language), a domain-specific language in the area of even processing,
    - o PMML (Predictive Model Markup Language)
    - o R scripts
- Ample data visualization capabilities

We would like to stress that all functionality of «Cumulocity IoT Core» is also exposed as REST services, thereby facilitating integration and application development on top of it.

Microservices & Application Hosting
Additionally, «Cumulocity IoT Core» shall provide the platform for hosting and enacting IoT-related or stand-alone **microservices** and, based upon these microservices and standard «Cumulocity IoT Core» functionalities, sophisticated **applications**.

Microservices and applications hosted on «Cumulocity IoT Core» are separated by Cumulocity's internal **«API Gateway»**.

Besides the required management capabilities like:

- **User Management**
- **Microservice Management** (using, amongst others, Kubernetes)
- **Application Management**

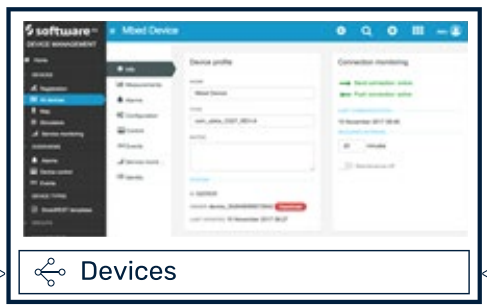
«Cumulocity IoT Core» incorporates the following unique and **differentiating features**

- **True multi-tenancy**
- **Subscription-based** microservices and applications
- Option to **deploy** the complete platform to **on-premise systems**

Devices

Device Lifecycle
- Device Inventory & Runtime Statistics
- Device Identity Management
- Credentials per individual device
- Provisioning for small & large deployment
- Auto-registration
- Asset management (network, location, ...)
- Gateway hierarchy and command routing
- Device Twin

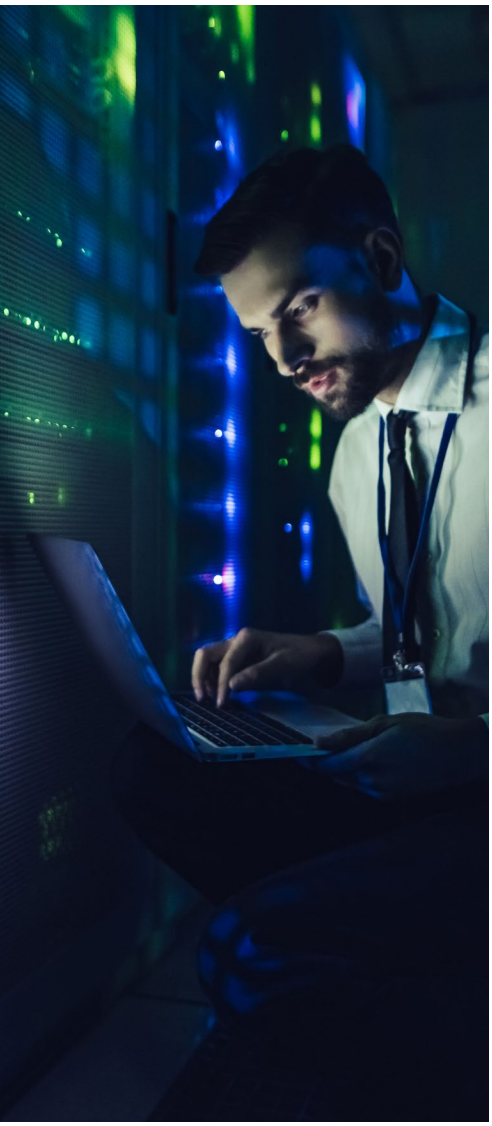
Connection Management
- Connection availability monitoring
- Connection metrics (RSSI, Signal strength)
- Switching between IP and SMS



Device Operations
- Firmware & software management
- Faulit & alarm management
- Configuration management
- Remote command execution
- Bulk operations with scheduling
- Troubleshooting: Remote shell, logs, ...
- Real-time alarms with integrated workflow

Remote Access
- Access screen of remote machine/ HMI
- Single sign login, per user access rights
- No shared password, no VPN, no cliend SW

all devices - all use cases - all networks



Main advantages:
1) Complete device management functionality including monitoring, alarm management, metrics collection and visualization, software and firmware management, interactive remote shell & screen sharing, configuration management, provisioning workflows, life cycle, identity management, connectivity platform integration.

Generic Cumulocity IoT platform features used by the device management application are: REST API access to all functionality, normalized device management data model, long-term data storage, multi-tenant support (incl. multi-level for resellers), role-based access control to support different user groups with different permissions, user audit log.

All features below are available in multi-language, brandable, responsive Web UI, plus via REST API:

Monitoring

- Connectivity monitoring
- Alarm management (workflow support, escalations, prioritisation, re-prioritisation, device SLA reporting)
- Metrics collection
- Visualisation

Diagnostics

- Interactive remote shell
- Connectivity platform integration (e.g. Jasper, COMARCH; for fast root cause analysis)
- Remote screen sharing (share device screen in web browser): For industrial IoT cases, customers need access to screens in case of support cases (e.g. share HMI screen)
- Ability to add diagnostics (e.g. thresholds) using Smart Rules
- Ability to add predictive maintenance using APAMA real-time analytics and/or Zementis prediction engine
- Event processing using power/business user configurable "Smart Rules" to create e.g. threshold alarms and alarm escalations.
- Customization of event processing rules using APAMA real-time analytics engine
- Device event logging
- Visualization of metrics, alarms, and connectivity.
- Configurable reporting

Configuration and control

- Device inventory (incl. filtering, manual groups, smart groups, map view)
- Configuration management (editing, repository, export, import)
- Device operations log
- Built-in device commands (restart, status upload, etc.)
- Device shell access

- Configuration for WAN and LAN settings
- Ability to add automatic fault resolution using APAMA real-time analytics (e.g. reboot device in case of certain error conditions)

Provisioning and authentication

- Device provisioning workflows (single & bulk, SMS support)
- Device life cycle support (e.g. device replacements, compromised devices)
- Device identity management (pre-device credentials, permissions)
- Multi-tenant provisioning support, e.g. managed service operator provisions device owned by customer later

Software & agent maintenance and updates

- Software and firmware management software and firmware repository
- Software/firmware approval workflow
- Multi-tenant provisioning support, i.e. managed service operator can schedule software updates for devices of customer, if customer has agreed for that; uses Data Broker feature to forward operations from operator tenant to customer tenant.

Event processing, reporting and visualization

- Event processing using power/business user configurable "Smart Rules" to create e.g. threshold alarms and alarm escalations
- Customization of event-processing rules using APAMA real-time analytics engine
- Device event logging
- Visualization of metrics, alarms, and connectivity
- Configurable Reporting

2) Optimizations of device management using Application Enablement platform features are: Cumulocity IoT Device Management is just one "App" on top of the Cumulocity IoT application enablement platform. Therefore all enablement features can be used in the context of device management, too:

- Perform automated root cause analysis and alarm correlation using APAMA real-time analytics, e.g. suppress temporary device problems
- Automate device management tasks using real-time analytics & event-driven actions, e.g. reboot device automatically in case of certain alarms
- Generate alarms based on collected metrics using Smart Rules, e.g. create alarms if mobile radio strength or battery is too low
- Extend the built-in device management user interface for device specific forms using Web UI Plugins. See http://www.cumulocity.com/guides/web/introduction/ for details
- Support new device types by using the Cumulocity IoT device SDKs for Java, C++, C#, Lua, etc.

3) Support the managed service business model: Cumulocity partners (e.g. telecommunication companies) plan to provide "Device Management as a Service" for customers using IoT. With Cumulocity, this is possible by sharing device management data with managed service organizations using Data Broker. This can be done on top without a need for any device or device protocol change. Customer stays in full control of all data.

4) Device management is a fully productized, documented, brandable, multi-language application. See http://www.cumulocity.com/guides/users-guide/device-management/ for details.

5) 100% device-, device vendor- and protocol-agnostic: Cumulocity IoT Device Management supports 100's of devices (incl. gateways) today, using different protocols (MQTT, REST, LWM2M, Proprietary, LoRa, Sigfox, NB-IoT, ...). New device types and protocols can be easily added. One foundation is the common device management data model, see http://www.cumulocity.com/guides/reference/device-management/ for details.

6) Simplified development: note that the "Cumulocity Edge Platform" is using the same software platform as the Cumulocity IoT platform in the cloud, so it shares the same features with respect to device management. Development efforts are eased with a single software architecture that allows the same APIs, protocols, data models and analytical models to be used from the edge to cloud, enabling centralized IoT solutions to efficiently evolve towards a distributed architecture as business requirements dictate.

Data management has been one of the most important areas of focus for development of the Cumulocity IoT Platform. This includes the following key capabilities and differentiators:

- Built-in IoT data model: The platform comes with a built-in IoT data model (device twin, alarms, measurements, events, ..) which includes predefined hierarchical models specifically implemented for industrial use cases. This is the key requirement to enable an App ecosystem on top of the platform (see global ADAMOS IIoT platform in tab "REF").

- Extensible: The data model is dynamically extensible for built-in data types to accommodate bespoke requirements. This was key to implement the 100+ customer solutions on top of the platform, see tab "REF.

- Fieldbus Connectivity with code-free approach: support for Brownfield Connectivity a number one requirement for IIoT platforms: Asset life cycle in IIoT ("machines") is 20+ years. These machines need to be connected in a code-free approach. The Cumulocity IoT platform Cloud Fieldbus feature provides this: a configuration-only, data-driven IIoT protocol (Fieldbus) integration, with model import and export capabilities.

- Native multi-level multi-tenant: the platform supports multi-level multi-tenant 100% data segregation. E.g. a reseller can host multiple smart device manufacture (SDM, each with its own tenant), and each SDM again can create for each of its customer (e.g. factory) an own tenant. We think that this is a number one requirement for all Industrial IoT cases, as factory owners caring about data privacy and security require 100% data segregation and isolation.

- Metadata-driven data normalization: different IIoT machines (even different machine versions from the same vendor) expose key data in different variables (tag/data point). With Cumulocity, these can be normalized to simplify application and analytics model development.

- Identical edge and cloud models: identical data models (and APIs) that span both edge and cloud to enable moving applications and analytics easily between cloud and edge deployment.

- Plug&Play device models: When connecting hundreds of different machine types (incl. different vendors, generations and versions) to an IIoT Platform, it is not possible to centrally pre-provision the machine data model to the platform. With our 20+ years of experience, this fails to scale once there are more than 50-100 machine types. Therefore Cumulocity IoT platform supports dynamic metadata flow from device to edge/cloud, plus automatic metadata generation based on instance data.

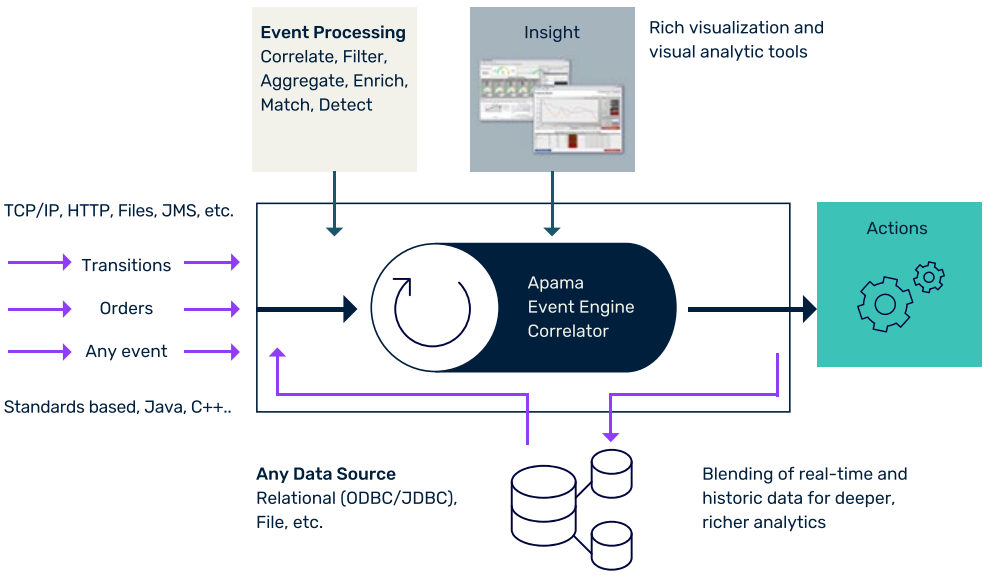
- We created an open, flexible solution which provides integration with data lakes such as Hadoop

- For high-speed analytics that draw on static as well as real-time data, Software AG provides an advanced, widely-used distributed in-memory store and compute engine (Terracotta)

Our Master Data Management (MDM) component provides capabilities for data governance, data management and data quality and is used for validating, enriching, standardizing, matching of metadata, master data and reference data, providing a single version of the truth for assets and rules on both streamed and reference/historic data.

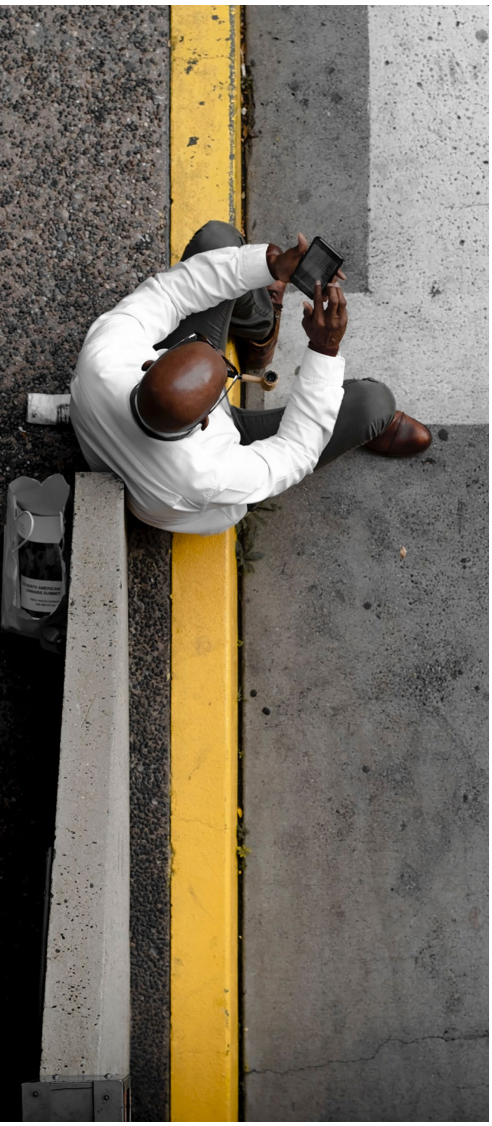
Real-time analytics with APAMA and Zementis

Apama Streaming Analytics (which complements our Cumulocity IoT platform) is Software AG's platform for analysing high-velocity streaming data to detect business & security risks and opportunities in real time. Zementis is our predictive analytics capability which works hands-on-hand with Apama to provide the complete and comprehensive streaming & predictive analytics capability.



It is the software that can filter, aggregate, enrich and analyze a high throughput of data from multiple disparate live data sources and in any data format to identify simple and complex patterns to visualize business in real time, detect urgent situations, and automate immediate actions.

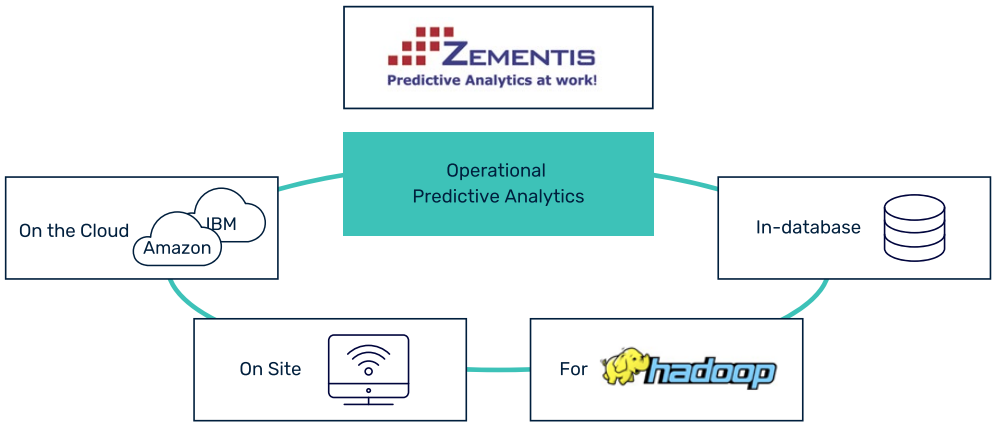
APAMA is designed to make sense of the huge volumes of events from social media networks that are generated each day. It allows organizations to identify sentiment and trends, then use this insight to automatically take action.

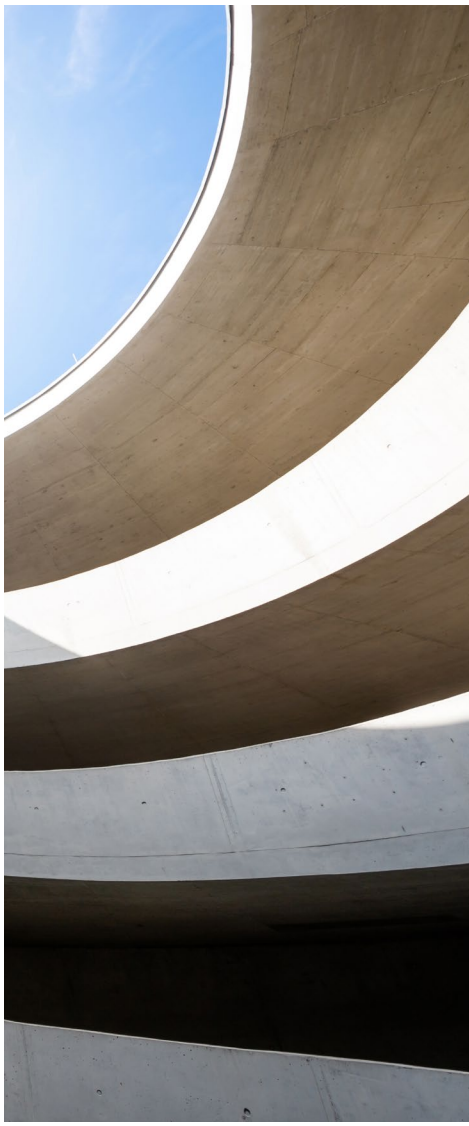


APAMA Key capabilities:

- Rich analytics – aggregations, temporal, filtering, and location
- Supports extreme scale and performance
- Blending of real-time and historic data for deeper, richer analytics
- Business level tooling
- Rich visualization and visual analytic tools
- In-memory architecture
- Complex event processing
- Support for predictive analytic models
- High performance messaging to mobile, web, IoT

Zementis has PMML Support; fully supporting the Predictive Model Markup Language (PMML) industry standard for data mining applications, addressing the highest execution requirements for batch processing, in-memory computation and streaming data.





With Zementis, count on a wide range of AI, machine learning and statistical algorithms for high-performance scoring, including:

- Association rules
- Decision trees for classification and regression
- Neural network models: back propagation, radial-basis function and deep learning
- Support vector machines for regression, binary and multi-class classification
- Linear and logistic regression (binary and multinomial)
- Naïve Bayes classifiers
- General and generalized linear models
- Cox regression models
- Rule set models (flat decision trees)
- Restricted Boltzmann machines
- Clustering models: distribution-based, center-cased and two-step clustering
- Scorecards (including support for reason codes and point allocation for categorical, continuous and complex attributes)

Multiple models: model composition, segmentation, chaining, cascading and ensemble, including Random Forest Models and Boosted Trees.

Zementis Predictive Analytics also implements the definition of a data dictionary, missing, outlier and invalid values handling, and myriad other functions for implementing data pre- and post-processing functions, including:

- Value mapping
- Discretization
- Normalization
- Scaling
- Logical and arithmetic operators
- Conditional logic (IF-THEN-ELSE)
- Built-in functions
- Lookup tables
- Business decisions and thresholds
- JSON™
- User-defined custom functions

ABOUT SOFTWARE AG GOVERNMENT SOLUTIONS

Software AG Government Solutions delivers leading edge software that helps the Government connect existing, new and future technologies together whether on premise or in the cloud. Leveraging our Government Integration Hub, webMethods, and our strategic IT portfolio platform, Alfabet, along with our highly effective "Prove IT First and Prove IT Fast" approach to solving mission critical IT challenges, we specialize in helping customers optimize large scale, mission critical solutions across complex extended enterprises. Learn more at www.SoftwareAGgov.com.

fs_cumulocity-iot_iot-platform_en

software AG
GOVERNMENT SOLUTIONS