Fusion Product Description

Version 2013R1

Table of Contents

1 Document Introduction

1.1 Document Purpose

The purpose of this document is to provide a high level description of the Fusion product; How it is deployed and integrated into the network, how it operates, which capabilities it offers and some examples of which problems it solves. Additionally, a high level overview of the architecture of the product is presented.

1.2 Document Audience

This document is written for those evaluating usage and deployment of provisioning systems for remote CPE management. Typically, the readers of this document are managers and system/network administrators of service providers offering broadband, VoIP, IPTV and other IP-based services.

1.3 Document History

Version	Editor	Date	Changes
2009R1	Fredrik Gratte	24-Feb-09	Initial public version.
2009R1-U1	Fredrik Gratte	23-Mar-09	Updated version.
2009R2	Fredrik Gratte	17-Jun-09	Added Solutions and Staging chapters.
2011R1	M. Simonsen	4-Jan-11	Added reports, removed OPP, added SPP
2012R1	M. Simonsen	9-Dec-11	Updated document to Fusion, 2012R1
2013R1	M. Simonsen	05-Mar-13	Updated document to Fusion 2013R1

1.4 Acronyms and Abbreviations

Acronym	Explanation
ACS	Auto Configuration Server.
APS	Automatic Provisioning System.
Fusion	Ping Communication's eXtensible APS with advanced features such as Service Windows, Job Control and Smart Groups.
Fusion Module	The Fusion consists of several independently running Modules. The Modules may run on separate hosts in the network.
North Side	Common term describing the part of the Fusion which includes all Fusion Modules that provide management interfaces.
South Side	Common term describing the part of the Fusion which includes all Fusion Modules that provide CPE communication protocols.
Core	Common term describing the part of the Fusion which includes all Fusion Modules that provide neither management interfaces nor CPE communication protocols.
CPE	Customer Premises Equipment. Used in this document to refer to a single physical device. Same as the term “Device”.

Virtual Provider	Service Provider of IP-based services to end-users. Typically offers services such as broadband access, VoIP and IPTV.
Hosting Provider	Service Provider of IP-based services to Virtual Providers. Typically offers a hosted solution including VoIP, IPTV and provisioning.
Native Deployment	Fusion deployment inside the network of a provider. Chosen by Providers who wish to run their own Fusion.
Hosted Deployment	Fusion deployment inside the network of a Hosting Provider. Typically chosen by Virtual Providers, who do not wish to run their own Fusion.
Fusion Administrator	Employee of a Hosting Provider managing network infrastructure. Typically the person deploying, configuring and running the Fusion.
Fusion Operator	Employee of a Hosting/Virtual Provider managing the provisioned deployment of CPEs through the Fusion Web Interface.
Fusion Data Model	How the Fusion determines which Parameter values and other changes to send to CPEs. Implemented as a database schema with additional processing logic on top in the various Fusion Modules.
Parameter	Each individual configuration setting is represented in the Fusion Data Model as a Parameter. A Parameter consists of a name and usually (but not always) a value.
Unit	A dataset in the Fusion database consisting of Parameter values relating to a single CPE. This dataset may extend beyond the Parameter values actually sent to the CPE, as some Parameter values may only be useful or needed by the Fusion itself. Also, the dataset may represent only a subset of all the configurable settings in the CPE. For these reasons, it is important to distinguish the term “Unit” from the terms “CPE” and “Device”.
Profile	Dataset stored in the Fusion containing Parameter values shared by multiple Units of the same Unit Type. A Unit is always assigned to a single profile. Multiple Profiles may be created for a Unit Type.
Unit Type	Units that represent CPEs of the same model share a common definition of that CPE model named Unit Type. The Unit Type definition is a list of Parameter names only, as the Unit Type never contains any Parameter values (values are stored in the Unit and/or Profile).
Group	A set of matching criteria used to search for Units. Commonly referred to as Smart Group.
Job	Automates and controls changes to Units within a Group. Partitions the changes over time according to rules to limit network load.
Job Chain	Multiple Jobs being automatically executed in a designated sequence.
Periodic Mode	Provisioning Mode where the Fusion automatically configures all

	CPEs based on their combined Unit and Profile parameter values.
Inspection Mode	Provisioning Mode where an Fusion Operator manually inspects and configures a single CPE through the Fusion Web Interface.
Staging	Fusion functionality used for optimizations in manufacturing, logistics and time-to-market for CPEs.
TR-069	Industry standard provisioning protocol used by the Fusion to read and write configurations from and to the CPEs, in addition to handle upgrades.
SPP	Simple Protocol Provisioning, covers HTTP(S)/TFTP
LAN	Local Area Network. Typically a network interface only reachable from the local network inside a customer premise.
WAN	Wide Area Network. Typically a network interface reachable from the Internet.
NA(P)T	Network Address (and Port) Translator. Application residing in network gateways translating between LAN address/port pairs and WAN address/port pairs.
API	Application Programming Interface. Used in this document to refer to interface(s) making it possible to integrate Fusion with systems from other vendors.

1.5 References

Document	
[1]	Fusion TR-069 Server User Manual
[2]	Fusion SPP Server User Manual
[3]	Fusion Syslog Server User Manual
[4]	Fusion Shell User Manual
[5]	Fusion Web User Manual
[6]	Fusion Database User Manual
[7]	Fusion Web Services User Manual
[8]	Fusion Core Server User Manual
[9]	Fusion Monitor Server User Manual

2 Introduction

Today’s IP-based technologies enable us to communicate with anyone, anywhere, at any time. As people become accustomed to the conveniences of IP-based technology, the market is expanding to accommodate new customers and faster, more advanced products. To stay ahead and ensure success, service providers must guarantee fast, reliable, and user friendly services to their customers while also keeping costs low. Additionally, they must be flexible enough to add new capabilities and services as they expand their offering. The Ping Communication's Fusion offers a solution for Providers who wish to increase customer satisfaction and decrease maintenance costs while also staying open to future service possibilities.

The Ping Communication Fusion enables Providers to remotely control and monitor customer premises equipment (CPE) and the services they deliver. Typically the CPEs deliver broadband internet access, VoIP and IPTV services to end users. Remotely controlled configuration as well as software management is necessary to ensure desired operation. Furthermore, problem detection through logging and service monitoring is required to detect and avoid degradation of the quality of service experienced by the end users.

The Fusion solution automates Provider provisioning tasks for any deployment size. Built as a massively scalable solution, handling thousands of devices per minute on even the most modest server hardware, the Fusion solution provides rich provisioning functionality across multiple device management protocols such as the industry standard TR-069 and more proprietary standards like HTTP(S)/TFTP. Scaling to deployments of millions of devices, the Fusion solution provides advanced features protecting the Providers network, the quality of services delivered as well as the CPE investment from disruption and damage through continuously controlled and monitored roll-out of changes.

3 Solutions

The Fusion provides solutions for manufacturers, logistics partners and service providers handling CPEs. Shortening time to market, simplifying manufacturing and logistics as well as automating and monitoring the CPE deployment are key benefits. Focus in the following chapter will be on the Service Provider solution, which is the most widely deployed solution.

3.1 Service Provider solutions

Fusion provides a sophisticated CPE remote management solution to service providers. Functionality such as smart groups, jobs and service windows ensure flexible, automated and continuously monitored roll-out of configuration changes and upgrades at configurable periods in time, protecting the CPE investment as well as avoiding service disruption for end users. The Fusion is light-weight, rapid to deploy and highly scalable, facilitating a scale-as-you-grow approach with low initial investment.

Additionally, features such as inspection mode provide remote access to each CPE in case such is needed by help-desk and support call-centres. Read on for further details on these features.

3.2 Manufacturer solutions

The Fusion provides a staging solution for CPE manufacturers which reduces time to market and simplifies both manufacturing and shipment. The solution ensures that service provider specific firmware can be prepared simultaneously with production and shipment, with the firmware being needed only at the time when the end-user powers up the CPE for the first time. Typically, a manufacturer deploys a single, global staging server to handle its devices.

Additionally, manufacturing is simplified because regardless of which service provider specific firmware the CPEs will run when going into service, they will all be manufactured with the same, pre-qualified and in-field upgradable firmware. The service provider specific firmwares are never involved in the manufacturing process, they are deployed only as in-field upgrades when the end users power up the CPEs for the first time.

3.3 Logistics partner solutions

The Fusion staging solution can also be used by logistics partners, which specialize in handling large numbers of smaller CPE orders. Such logistics partners typically function as CPE suppliers for smaller service providers, or service providers in an early phase ordering smaller CPE volumes.

The logistics partner typically orders larger quantities of CPEs from the manufacturer. When a service provider orders a small quantity of CPEs, the logistics partner ships the CPEs (typically directly to end users), simultaneously informing the Fusion Staging server which firmware to apply to the CPEs being shipped. The actual upgrade does not take place until the end users power up the CPEs for the first time.

4 Deployment

The Fusion is designed to be rapid and easy to deploy. Experience shows that a single network administrator deploys and configures the Fusion to up-and-running in a few hours. Additionally, the Fusion offers multiple convenient ways to load the system with the necessary CPE configuration settings.

This section will briefly mention platform and network requirements, as well as details on Native Deployment and Hosted Deployment.

4.1 Platforms

The Fusion Modules are mostly Java implementations built to run on a standard JEE middleware, meaning on top of any OS with a JEE implementation running on it. We recommend running on Ubuntu 12.04 LTS.

The Fusion Modules is dependent upon MySQL 5.5 for database. The Fusion Database schema is loaded into one of these database servers as part of the installation procedure. The database server itself is not part of the Fusion product, and must be appropriately installed and configured by the Provider prior to installing the Fusion.

4.2 Network and interfaces

The Fusion is deployed inside the corporate network infrastructure of a Provider. Typically, the Fusion has its South Side Modules (more on architecture below) exposed to the public internet, for CPEs to interface with. The Core Modules of the Fusion should always be located on the Providers secure and private corporate network, inaccessible from the public internet. The North Side Modules are frequently only accessible from within the corporate network, however interfaces such as the Fusion Web interface and the Fusion Web Services may be exposed for external access from the public internet, as long as they are appropriately secured against unauthorized access.

4.3 Native deployment

A Native Deployment refers to when the Fusion is deployed inside the Providers own network infrastructure. The Provider runs its primary services (broadband internet access, VoIP, IPTV etc.), in addition to the Fusion which is used to remotely control the CPEs delivering the primary services to the end users.

A native deployment means that the Provider assumes the roles of being both the Fusion Administrator as well as the Fusion Operator.

4.4 Hosted deployment

Increasingly the Fusion is being deployed as part of a hosted solution offered by Hosting Providers. Such a solution often includes VoIP, IPTV, provisioning and possibly also other services targeted at Virtual Providers.

A Hosted Deployment means that the Hosting Provider assumes the role of being the Fusion Administrator, while the Virtual Provider is left only with the role of being the Fusion Operator. This reduces the requirements for both technical skills and man-hours on the Fusion Operator, since the Fusion Administrator handles the more complex tasks (scalability, database backups, load-balancing etc.). The trade-off is that the Fusion Operator gives up some configuration flexibility of the deployment compared to a "Native deployment".

The Fusion implements a concept named Containers, which enables the Hosting Provider to use a single Fusion deployment to serve any number of Virtual Providers. This means the Hosting Provider can take on an increasing number of Virtual Providers as customers, without being burdened with increased administration overhead of the Fusion. Furthermore, the Fusion scales equally well regardless of hosting many Virtual Providers with few CPEs each, or hosting few Virtual Providers with many CPEs each. It is solely the total number of CPEs that is relevant to performance and hardware requirements.

The Virtual Providers and all their private content stored in the Fusion (including CPE logs and configurations) are completely isolated from each other, thereby the name "container". Each Virtual Provider "lives" in a container, without any possibility of seeing or be seen by the other Virtual Providers using the same Fusion deployment. Only the Hosting Provider has full access to all contents in the Fusion database.

5 Fusion in birds view

This chapter aims at a relatively short description of what the overall capabilities of Fusion are.

5.1 Summary

Fusion is a provisioning and monitoring system for CPEs supporting TR-069 and some other protocols. We believe the strong points are some features for monitoring combined with the ability to act upon certain events. The actions are themselves very flexible. Furthermore, Fusion is a light-weight and easy-to-install system. We believe one of our strong points is the flexibility in terms of change requests from customers – and still maintain a generic system, not one that is custom made for one customer only. Performance is another important issue, since a rigorous system architecture tend to be bloated over the years, hindering great performance. Fusion has avoided all that, always been kept lean and is therefore possible to scale to really large environments. The pricing is also quite attractive, since the organization behind the product is also quite slim and efficient.

5.2 Provisioning

Fusion is a provisioning system, mainly based on the provisioning protocol TR-069. However, it can also provision devices using TFTP or HTTP(S). This makes for a system which can handle a large variety of CPEs in the market. Fusion is also capable of discovering and auto-registering new devices. We believe that with these capabilities it should be fairly easy to get started quickly. Within provisioning itself, there is a wide array of features. One way of viewing this is “Simple provisioning” versus “Advanced provisioning”. The idea is that both modus operandi is supported. In the advanced modus you can set up certain chain of provisioning events and to control the outcome and the success of the action. Furthermore, Fusion tries to make the underlying provisioning protocol as invisible as possible, so that you may focus on the device and not the technicalities/limitations of the protocol.

5.2.1 TR-069

This is the back-bone of the system, since most modern CPEs these days support TR-069. Many of the devices still has oddities, bugs and quirks, which makes it paramount for the server to be flexible with the devices. The TR-069 server offers some special quirks which can be turned on/off depending on the device connecting to the system. The server itself is designed to be accommodating with regards to the device input. On top of this, Fusion offers a fantastic capability in terms of testing.

5.2.2 HTTP/TFTP

These servers are rather simple, since the protocols don't offer much provisioning-vise. Basically, the server can serve a configuration file to the devices, containing all the necessary data. This file format is not standardized, and for this reason, every new device must go through a quick development phase.

5.2.3 Telnet

Telnet is not a provisioning protocol, but it is a very powerful interface to the devices. Since almost all devices support this, and since a lot of information and possible actions are only available over Telnet, Fusion offers the ability to execute Telnet scripts to a set of devices. With these scripts, it possible both to execute commands on the device AND read data from the device. This is probably a unique feature among provisioning server systems.

5.3 Monitoring

5.3.1 Syslog

A speciality of Fusion is the emphasis we put on syslog. This unsecure and unreliable protocol is actually very valuable, since most devices today offer some kind of syslogging to a remote server. Instead of trying to force the device to support a rigorous set of TR-069 parameters (the data models from Broadband Forum are indeed voluminous), we instead try to use the data coming from the devices already and harness it's information value. Therefore, Fusion offers a syslog server which enables it to collect data from the devices and store it in the Fusion database. This information can provide extremely helpful if error occurs in the device. To further enhance the monitoring capabilities of the device, Fusion offers a standard set of messages which the device may support – to signal things like uptime/downtime on Voip-services, Voip-call summary, internet connection uptime/speed, etc. These things are of course not mandatory, but can be an extra treat to Fusion.

The following sub chapters outline some of the features built on top of syslog.

5.3.2 Reports

The most basic report is the syslog report, simply counting the occurrence of errors/notice/warning/etc for each device type. You can choose the see the "executive overview" or you can zoom into the details, to identify which device(s) are responsible for the messages. The reports are meant to get an overview of the situation, to see the "big picture"; trends over weeks and months.

There is a number of reports, depending on the device your using.

5.3.3 Syslog Event/Trigger

You can design certain actions based on the syslog messages from the devices. It's possible to remotely reboot a device because a certain message indicate an error condition. This topic is covered later as well.

5.3.4 Heartbeat detection

To be sure, a CPE is a complex device these days. There can be a number of internal services running on a CPE. If such a service is emitting a so-called "heartbeat" every hour or every X hours, then Fusion can detect the absence of such a message, and react to that.

5.4 Interfaces

5.4.1 Web

The main interface of Fusion is the Fusion Web. This allows a user to perform almost any action possible within Fusion. It's a great tool to view reports and to perform simple analysis of the traffic patterns, etc. You can easily set up complex jobs and provisioning schedules. If you want to do repetitive task, you should look to the Shell.

5.4.2 Shell

A professional system is nothing if not supported by a decent CLI (Command Line Interface). With this you can automate and perform large operations if needs be. Complex searches for special units, combined with actions on those device can be done fairly easy, using a one-liner command. The Shell interface resembles to some extent a regular Unix shell, but also has it's own specialities. The Shell comes with an in-built help system to make it easy for you to learn while typing.

5.4.3 Web Services

To allow for machine-to-machine integration, Web Services are offered. This services makes it possible to query, add, change and delete the most basic concepts of Fusion. These services are very well documented and have been used in several customer cases. To this day, these services have provided the expected level of interoperability.

6 Fusion feature details

The Fusion is designed as a true multi-protocol provisioning system, ensuring that Fusion features described here will work across most protocols. For the currently implemented protocols (TR-069, HTTP(S), TFTP, Telnet), all features described here.

The Fusion architecture places as little requirements on the CPEs and provisioning protocols as possible (read more Fusion architecture below), but some provisioning protocols (which might be added to Fusion in the future) may not be able to send even the most basic information back to the Fusion, and thus some Fusion features may not be possible in combination with certain protocols.

6.1 Search

As the numbers of Units in the Fusion grow, the search capability is increasingly important. Not only to find individual units based on MAC id, Unit id, customer number or any other unique identifier, but also to find sets of devices which has a common behaviour. The various properties of the CPEs can be customized and listed in a summary page to get a quick overview of which devices should be paid special attention to. The list can be sorted by each column.

When choosing advanced search you can make searches like:

1. Parameter1 interpreted as number must be larger than 100 AND Parameter1 interpreted as number must be lower or equals to 1000.
2. Parameter1 interpreted as text must differ from 10.100 followed by anything AND Parameter2 interpreted as number must not be 178

This allows you to specify very detailed searches, to narrow down the list of CPEs to investigate.

6.2 Service Windows

Service Windows can be optionally configured by the Fusion Operator to ensure that certain changes to the CPEs are performed only at appropriate moments in time. Depending on the services delivered by the CPE in question, certain changes such as software upgrades are not appropriate within certain parts of the day (or week). There are different types of Service Windows according to the disruption (or non-disruption) the changes are expected to cause to the services delivered by the CPE.

Common use-cases are IPTV devices that should not have any software upgrades performed during prime-time television hours. Or for instance VoIP devices that should not see any configuration changes during work-hours.

The Service Windows functionality is tightly integrated with load spreading mechanisms, ensuring that the load from the deployed pool of CPEs is spread out within the Service Windows (and not outside them). The Jobs functionality also respects the Service

Windows, and will request CPEs to re-connect within the appropriate, next Service Window according to the type of Job that is to be executed.

6.3 Smart Groups

The Fusion architecture supports gathering units into so-called Smart Groups by defining rules used to match against the Parameter values of the individual units. Thus, units are not statically attached to a group (like they are to a Profile), instead they drop in and out of groups according to changes in their Parameter values. As the definition of a Smart Group is actually just a search, having Units dropping in and out of groups do not cause any updates to the database.

Creating, modifying and destroying groups are therefore a near-zero effort for the system in terms of load. No actual searches are performed in the database when setting up such groups (unless to give indications of the size of the group etc.). Instead, units are matched against the group definitions when the units are being provisioned, or when an Fusion Operator requests to see a list of units within a group. Thus, whether the group has a single or millions of units, this does not affect performance when manipulating the group definitions.

The search capability of a group is the same as for the search capability for Units in the advanced mode.

Groups can be structured hierarchically, and the hierarchy can be rearranged at any time. The system scales to large numbers of groups, with minimal impact on provisioning performance.

6.4 Jobs

The fundamental idea behind Jobs is to automate tasks that would alternatively be executed manually by an Fusion Operator through an Fusion management interface. This is not as easy as it sounds, as the advantage of the attention of a human is that problems can be manually detected, and further tasks halted immediately to protect the deployment from further damage or the service delivered from disruption. Automating tasks is not an option if it involves the risk of turning hundreds of thousands of CPEs into little more than bricks, or equally many angry end-users.

Experience demonstrates that one can attempt to imagine and simulate all sorts of potential problems in a lab environment, however the actual roll-out of changes to the CPEs in a production deployment will always involve unknown variables and problems, potentially damaging a CPE deployment and possibly introducing the enormous cost of replacing perhaps hundreds of thousands of CPEs.

6.4.1 Job types

The most typical job to run is a CONFIG job which simply changes the configuration on a device. Another is SOFTWARE and SCRIPT job, to upgrade/downgrade firmware or to run a proprietary script on the device. If you're dealing with a troublesome device, you might consider the RESTART or RESET jobs. These jobs will either reboot the device or perform factory reset.

A KICK job is another very useful job type, if there is an urgent need to provision a set of changes quickly. Instead of waiting until the devices connect to Fusion, you can have Fusion kick start the changes almost immediately.

You can also mix regular TR-069 and Telnet provisioning (or do only Telnet), using the TELNET job. The job will immediately connect to the devices and run the Telnet-script attached to the job. But you can also configure Fusion to retrieve information from the Telnet-script run.

Perhaps the most advanced provisioning type is to run a SHELL job. As the device connects and provision, a customized shell script attached to the job is triggered. In the script you can perform any kind of logic on the Fusion system, to perform more complex provisioning changes than possible using just CONFIG type. One example is the ability to change ACS-username and create a new Unit Id in the system upon connect of the device. This is useful if the device has been produced with non-unique ACS-usernames.

6.4.2 Controlled and un-controlled changes

An Fusion Operator can make changes to CPEs without the use of Jobs, by updating the Unit (when updating a single CPE) or Profile (when updating multiple CPEs at once) instead. However, and especially when updating values in a Profile (which can affect thousands of CPEs), the change is un-controlled, and may therefore potentially cause unintended harm to large amounts of CPEs. When changing for instance a Profile, the Fusion will perform the change to any CPE belonging to that particular Profile, without checking if the change actually caused damage to the CPEs.

Ping Communication has carefully designed the Jobs functionality to continuously control and monitor the roll-out of changes in production deployments. Each CPE that is being changed somehow (Parameter change, software upgrade etc.) is being carefully watched over to ensure that the change did not turn the device into a brick. If any problem is detected, the Fusion Core Server will immediately stop the Job so that the problem can be manually inspected and resolved before it is allowed to cause further damage. A Job can subsequently be modified (if desirable) and started again if/when appropriate.

The Job functionality is designed to handle large numbers of Jobs without significant impact on performance.

6.4.3 Jobs and Smart Groups

A Job is always tied to a Group of units on which to execute the Job. If the Group has children, the Job will be executed on the children as well. The Job will not be executed on any unit which do not match the Group definitions. Additionally, the Fusion carefully checks the software version of the CPE before executing a Job, to ensure that the Job is executed only on a software version that is known to handle the Job.

6.4.4 Execution and Verification Stages

The actual change is performed in two distinct stages, named Execution Stage and Verification Stage. The former performs the change, while the latter verifies that the change was actually applied and that the CPE did not fail somehow. Also, to ensure that the Fusion database contains Parameter values consistent with those residing in the CPEs, the changes are not applied to the CPE's corresponding Unit in the database until the Verification Stage succeeds.

6.4.5 Configuring Jobs

Jobs can be configured by the Fusion Operator, specifying rules instructing the system under which circumstances the Fusion Core Server should automatically stop the Job. In addition to detecting different types of failure situations, rules can for instance also specify to start the Job by only changing a limited number of CPEs, allowing the change to be manually inspected by humans before proceeding with the roll-out to the entire CPE deployment. There are also other rules, for instance (but not limited) to control the speed at which changes are rolled out, which avoid network congestion during software upgrade Jobs.

6.4.6 Job Chains

A Job can depend on a another Job to be executed first. This ensures that multiple changes can be scheduled for CPEs, and these changes will always be executed in the designated order. This is useful for instance when a unit could need upgrading before a certain change to the Parameter values can be applied. As another example, manufacturers frequently only test upgrades from version A to B and from version B to C before shipping, not necessarily upgrades from A to C (without intermediate upgrade to B). Therefore, the safest way of upgrading is often to go through all intermediate software versions, as has been tested by the manufacturer.

Jobs can also be defined as "Infinite", meaning that they can remain in the system indefinitely, and execute whenever a CPE matches the Group and Job requirements. Multiple such "Infinite" Jobs forming Job Chains (using the Job dependency feature) can significantly automate provisioning efforts for Providers deploying a certain CPE model over time.

Commonly, the CPEs will over time arrive at the customers premises with differing software versions preloaded from the factory. Depending on which software version running in the CPE, different tasks could be needed (such as different upgrade-paths, different Parameter changes etc) depending on problems specific to each software version. The Fusion Operator can specify all changes needed for a particular software version (including the upgrade to the next version) as a Job Chain of "Infinite" Jobs.

The Providers will always experience CPEs running any historical software version connecting to their provisioning system. This happens due to manufacturers changing the software version preloaded into the CPE from production, and the fact that some CPEs can be left unconnected (by the end-users) for up to several years, and thus do not follow the changes rolled out by the Provider over time.

Thus, it is beneficial that regardless of software version the CPE is running, it will when connecting to the Fusion always drop into the correct Job Chain to execute (which could have been configured years ago by Fusion Operators), and in correct sequence go through all the historical changes Fusion Operators have configured in the Fusion for the type of CPE in question. Thus, there is no risk of applying certain Parameter value changes on the wrong software version, or risking an (untested) upgrade directly from version A to C. The Job Chains can also be modified along the way, for instance if the Fusion Operator has verified that a CPE model can be upgraded directly from software version A to C.

Job Chains of "Infinite" Jobs ensure that historical efforts (configuration changes, upgrades etc.) done by Fusion Operators configuring the CPE deployment will be

automatically reused by all subsequently connecting CPEs of that type, according to their currently running software version.

6.5 Reports

One might view a provisioning system as gold mine. It is in fact a system which controls a vast number of devices in a large network, and this is in itself a precious resource. Fusion has chosen to focus on two primary data sources; the statistics of the devices found in the Fusion database and the syslog data sent from the devices.

With this information Fusion can provide a magnitude of charts/reports. These reports can be inspected with regards to long time spans (years) or very short time spans (hours). A report can also be viewed from various "angles" (various metrics). These reports aim to provide an Fusion Operator/Administrator an overview over things like software version distribution, upgrade speed, failures, units connectivity with Fusion, etc. With syslog reports from the devices, even more sophisticated reports are available, Fusion can report the service level of VoIP for each device, and even start from top-level and find the overall service level of VoIP, then zoom in on a particular day/hour and then identify a set of units which had placed calls in that time period. This provides a standard way of going top-down to identify a particular issue.

6.6 Dashboard

The dashboard provides help-desk people to get a quick overview of the device in question. The dashboard shows a speedometer which sums all the important factors of the well-being of the device. The factors themselves are also shown, and they are coloured according to the state they're in. Furthermore, the dashboard offers a history, to see previous voice calls, boot history, memory statistics and syslog messages. All in all, the dashboard is a source to determine causes of issues and to troubleshoot the device. The reports mentioned in the previous chapter ties in with this dashboard, because one can drill down from the report all the way down to each individual dashboard. Together these features offers a superb control of the quality of the services experienced by the customers.

6.7 Logs

6.7.1 Data collection

Fusion has built an "ecosystem" around syslog, which is a de-facto standard for logging from CPEs. Fusion makes it possible to collect all messages from the CPEs and connect the messages to the correct Unit in Fusion. Furthermore, all modules in Fusion may log to the database, which will give a complete picture of all events connected to a CPE (changes in Fusion Web, Shell, provisioning from server, device logs, etc).

6.7.2 Filtering and analysis

Fusion offers powerful tools (both in Web and Shell) to search for specific logs. It might be that only messages coming from one specific module or device-type, containing a special string and has a given severity level, is of interest. This kind of searches as lots of others are available.

6.7.3 Syslog events

A syslog event is an action that Fusion Syslog Server performs upon receiving a message from a device. This action could be to suppress duplicate messages, to drop certain types of messages or to run a script on the device triggering the message. This makes it possible, both to trim the amount of messages stored, and to act upon certain events in the device. One action could be to set the device in DEBUG-mode, and thus receive even more messages.

7 Staging

The Fusion solution implements staging functionality to optimize manufacturing, logistics and time to market for CPEs. The advantages of a staging solution is best described by presenting a few, well-known problems in the CPE manufacturing business.

7.1.1 Time to market and logistics problems

The first problem is time to market. Preparing a firmware according to Service Provider specifications takes a certain amount of time, and so does the subsequent manufacturing and shipment, all the way from production facilities to the end-user itself. Each of these two tasks can easily take 2-3 months, resulting in a 4-6 months delay from the time of placing the purchase order to the moment the end-user connects the CPE and activates the service. This delay could be drastically reduced if the two tasks could be executed simultaneously.

The second problem is the shipment itself. CPEs manufactured with firmwares especially made for a specific service provider or region may end up being shipped to the wrong service provider, or simply the wrong region of the world. If such stranded CPEs could be easily used by another service provider (perhaps in another country with different firmware requirements), this would introduce significant savings and/or reduced risk.

A third problem is the large number of smaller service providers, each possibly with slightly different requirements to the CPE firmware. These service providers typically orders small batches frequently, as they grow their customer bases. Handling many firmware variations and small production batches in production lines is resource consuming, and it would be better if the all CPEs could be manufactured and shipped to a logistics partner with the same firmware. Subsequently, the logistics partner could handle all the small volume shipments to service providers, assigning the correct firmware for the CPEs of the shipment in question, at the time the shipment is effectuated.

7.1.2 The Staging remedy

The Fusion Staging solution is targeting these problems. The CPE production can be started immediately upon placement of purchase order, by manufacturing the CPEs with a minimal and in-field upgradable firmware. This firmware is not the one desired by the service provider, but rather a stripped-down version pre-qualified by the manufacturer. While the manufacturing and shipment takes place, the firmware desired by the service provider is prepared and qualified by developers.

At the time the CPE arrives with the end user, the CPE is plugged in and powered up for the first time. The minimal firmware in the device connects to the manufacturers Fusion Staging server. At this point in time, the developers have finished the service provider specific firmware, and loaded it into the Fusion Staging server. Also, the serial numbers for the CPEs shipped for the service provider in question have also been loaded into the Fusion Staging server, and mapped to the service provider specific firmware. When the server is contacted by the CPEs, the server locates each CPE in the Fusion Staging server database, and upgrades the CPE to the firmware it is mapped to.

Subsequently, the CPE will typically reboot into the service provider specific firmware, and connect to the service providers own management servers. Last, the service is

activated for the end users CPE (by the service providers management server). The Fusion Staging server never communicates with the CPE again.

Should the CPE by accident have been sent to for instance the wrong geographical region, the shipment of CPEs could be sold to another, local service provider, even though that local service provider have other requirements to the CPE firmware. The serial numbers for the shipment in question could then be re-mapped in the Fusion Staging server to another firmware (specified by the local service provider), and the CPEs could still be distributed to end users in the region the CPEs ended up.

Occasionally, manufacturers and service providers will use logistics partners to handle the many smaller volume CPE orders, and also the variations over firmwares between the service providers. Either the manufacturer or the logistics partner will then typically deploy an Fusion Staging server, and the logistics partner will configure each shipment (mapping the CPEs in question to the appropriate firmware) through the Fusion Web interface in the Fusion Staging server.

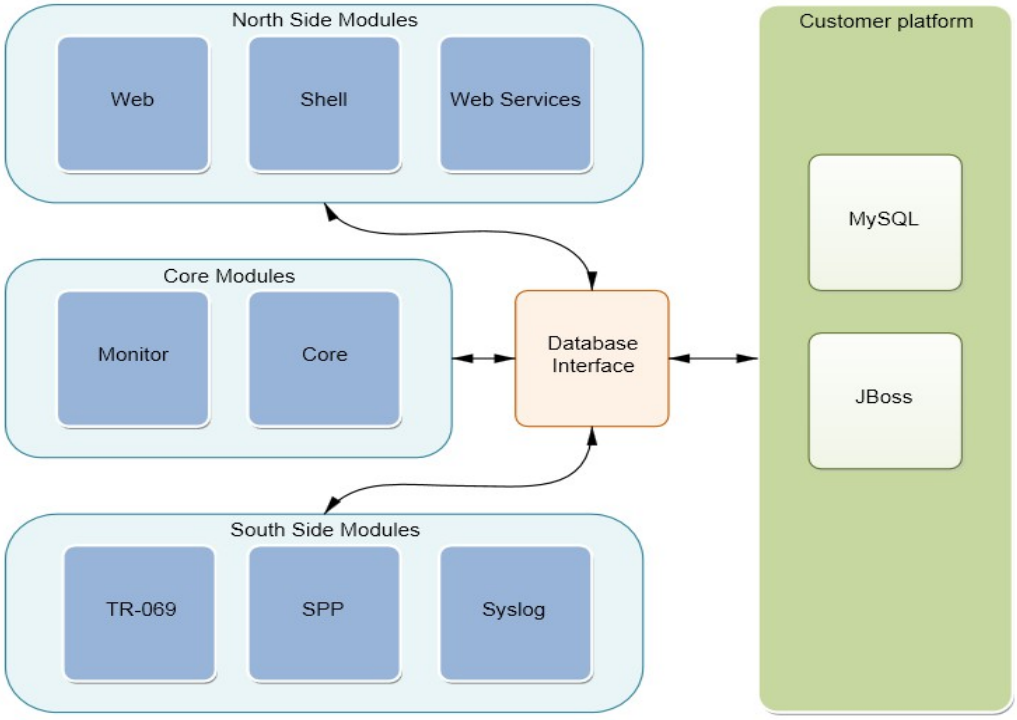
7.1.3 Staging operation

Each manufacturer typically deploys a single, global Fusion Staging server, which all manufactured CPEs (with the minimal firmware) are hard coded to connect to at power-up. The manufacturer administrates and operates the Fusion Staging server, as well as the platform it runs on. Alternatively, Ping Communication offers a hosted Fusion Staging solution for manufacturers using the Ping Communication CPE solutions platform. In this case, Ping Communication is handling the Fusion Staging server administration, while the individual shipments are configured by the manufacturer.

Currently, the Fusion Staging functionality is accessed through the Fusion Web interface. Through this interface, lists of CPE serial numbers from the manufacturer in addition to firmwares can be loaded into the Fusion Staging server database. In the future, the staging functionality is planned to be able to integrate to external, third-party systems to dynamically decide target firmware for the upgrade based on arbitrary and configurable criteria, typically some sort of geo-location data.

8 On the inside of Fusion

The Fusion is a highly modular architecture, with independently operating Modules all communicating through a central database. All Modules are implemented in Java and built to run on any JEE compliant middleware. This provides virtually unlimited possibilities for distributed deployments across networks, hardware platforms and operating systems. Below is shown an illustration of the Fusion architecture:



Fusion is installed on the Providers own hardware and software platform, and requires the Provider to prepare a JEE middleware stack (such as JBoss) as well as a database server (MySQL) prior to the Fusion installation.

Only the South Side Modules need exposure to the Internet, as these are the only ones communicating with the CPEs. The North Side Modules typically only need access from within the Providers corporate network. Last, the Core Modules do not have any other communication interfaces than the client connections they initiate towards the database server.

The individual Fusion Modules communicate between themselves exclusively through the database, there is no additional message bus to consider when deploying the Fusion into a network.

8.1 North Side

North Side Modules implement management interfaces to the Fusion. Fusion Operators manage the deployments of CPEs through these interfaces. The north side Modules offered are:

- Fusion Web – A web interface for browser-based management of CPEs.
- Fusion Web Services – a SOAP/WSDL API for integration with 3rd party infrastructure.
- Fusion Shell – A high-level scripting interface to management of CPEs.

These Modules are introduced in subsequent chapters of this document. They provide partially overlapping and partially complementary management capabilities, with interfaces suitable for a wide variety of tasks and integration needs.

These modules are designed to easily encapsulate more features and additional APIs for the future, while maintaining API stability towards customer interfaces where relevant.

8.2 Core

Core Modules are Modules that do not expose interfaces to external applications or networks. These Modules interface only with the Fusion database, and act according to changes and requests inserted there by other Modules. The core Modules are:

- Fusion Core Server – Advanced Job control, report generation, syslog trim
- Fusion Monitor Server – A monitor verifying the correct operation of other Fusion Modules

Additionally, all Modules make use of the Fusion Database Interfaces, which provide database vendor abstraction and additional, Fusion specific logic for handling Parameters. The Fusion makes use of plain SQL language queries to the greatest extent possible, to facilitate support of additional database server vendors in the future.

The Core Modules are completely agnostic to properties such as what provisioning protocol the CPEs are managed with, what the individual Parameters stored in the database mean, or where they come from (through which management interface). The core Modules handle automation and progress monitoring of tasks, surveillance of the server Modules and other maintenance tasks. Core Modules exist primarily to relieve the Fusion Operators of having to manually monitor and perform changes to the deployment of CPEs.

8.3 South Side

The South Side Modules are responsible for the communication with the CPEs. These Modules ensure that Fusion features discussed in this document work across protocols (as far as possible for the protocols in question). Currently supported management protocols are:

- Fusion TR-069 Server – Industry standard provisioning protocol.
- Fusion SPP Server – Simple Protocol Provisioning Server, handles HTTP(S)/TFTP/Telnet

Additionally, the Fusion Syslog Server (a syslog message collector) is considered a South Side Module, since it exposes an interface to the CPEs. There Fusion Syslog Server is not a provisioning Module, but an industry standard interface for CPEs to send their log messages to.

The South Side Modules are implemented with focus on stability, performance, scalability and simplicity. They must scale extremely well to handle even the largest CPE deployments imaginable in the years to come, and should be capable of handling large deployments even with modest hardware. These Modules are connected to the public internet so that they can be reached by the CPEs.

8.4 CPE considerations

Most service providers have mixed experiences with CPE devices. Historically, many CPE models have been buggy and troublesome pieces of equipment to work with. The "features vs. time-to-market" conflict during development cycles of such devices frequently neglect proper testing and stabilization efforts before manufacturing and shipping.

The Fusion is carefully designed to provide key provisioning features to Providers, while still requiring as little as possible from the CPEs. For instance, just a few of the methods in TR-069 is used, and the Fusion attempts to minimize the amount of data sent to and requested from CPEs at any given time. This to help avoid the CPEs from running out of memory and crashing (which is especially common for TR-069 enabled devices). The Fusion architecture therefore strive to implement and resolve as many of the provisioning features on the server side as possible, instead of attempting to rely on the CPEs to be confirming to for instance the TR-069 (and related) specifications.

As a few examples, the Jobs feature requires merely that CPEs can inform which software version and Parameter values they are using, the rest is handled server side. And Service Windows require only that the CPEs can be told to reconnect in X seconds from now (the CPEs do not need knowledge of wall-clock time), all other calculations and logic is handled exclusively on the server side. Yet other examples are the Periodic and Inspection provisioning modes, as well as the Smart Groups and Profiles, which are all server-side features only. The CPEs do not have any knowledge of such concepts, nor do they need to store or provide information to support these concepts.

8.5 Robustness

There are many aspects that affect robustness, and this chapter will only briefly mention some of the most important design aspects. For more detail on Fusion operation, please refer to [10].

8.5.1 Transactional storage

The Fusion is using transactional database storage and a database schema with extensive usage of foreign keys to help ensure integrity of the data in the database.

Additionally, care has been taken to decide when and in which sequence a south side Module may write changes to the database, ensuring that the CPEs and the Fusion database to the greatest extent possible always contain the same values. The system has been designed to gracefully handle disruption in database server access at any given time, being able to continue operation without any Fusion Administrator or Fusion Operator intervention once the database server is back on line. This is particularly important for instance during execution and verification of Jobs.

In certain situations, a change to the database requires writes in more than a single row or more than a single table for the data integrity to be preserved. The Fusion Modules take care to include such multiple writes into a single database transaction, so that changes written are always "either all or nothing". The Modules will on success commit the transactions, and on failures ensure a roll-back of the entire transaction attempted.

8.5.2 Concurrent connections and spread

The south side Modules have configurable settings for the maximum number of concurrent connections they are allowed to handle. Subsequent incoming connections from CPEs will be rapidly rejected to release operating system resources as soon as possible.

Additionally, the South Side modules implement load-spreading mechanisms to attempt to partition the load on the servers and network over time.

8.6 Scalability

There are many aspects that affect scalability, and this chapter will only briefly mention some of the most important design aspects. For more detail on Fusion operation, please refer to [10].

8.6.1 Database usage

The Fusion has been carefully designed to induce as little load as possible on the database servers. All functionality has been designed with scalability in mind, by ensuring that unnecessary transactions do not take place. The amount of data transferred for each transaction is kept to an absolute minimum, and the database schema ensures that all queries performed, especially those by the South Side Modules, always hit indexes in the database server. Still, the number of indexes is kept low, ensuring fast insertion and update queries.

8.6.2 Separate syslog database

Depending on the number of CPEs logging to the Syslog server, and the configured verbosity of the CPEs, the amount of messages collected may introduce a substantial load on the database. Therefore, the Fusion Modules can optionally be configured to use a separate database for the syslog messages. Regardless, Database administrators are advised to refer to the database vendor documentation, to ensure that the database server is configured appropriately for the loads expected.

8.6.3 Load-balancing and fail-over

Separately, and depending on implementation, provisioning protocols residing in CPEs may have additional scalability mechanisms built-in. One example could be devices using server-side configurable DNS SRV records for load-balancing and fail-over. Devices usually have back-off mechanisms for high load and avalanche situations. Such avalanche situations may occur when power is restored for large numbers of CPEs simultaneously, after power-outages in larger city areas. Please refer to [2] for more details.

8.6.4 Multiple Module instances

The Fusion Modules are implemented so that they can be deployed as multiple instances (located across multiple hosts) working in parallel. If any state information is needed across provisioning sessions, this information is stored in the central database so that the next provisioning session for a given CPE will find it regardless of which instance of the server Module was handling the CPE at the previous provisioning session. This is particularly important during the execution and verification of Jobs.

The Fusion Syslog Server can be installed as multiple instances as well, but syslog agents (typically residing in the CPEs) are virtually always configured with an IP address for where to send the syslog messages. Thus, an external load-balancer in front of these Fusion Syslog Server instances would be necessary. However this should only be necessary in case of very large deployments with very verbose CPEs, as the Fusion Syslog Server has several built-in mechanisms to handle peak loads and large volumes of messages. It is highly likely that the database-server will become the bottleneck long before any of the Fusion Modules will. Read more about this Module in [3].

8.6.5 Caching mechanisms

The Fusion Data Model allows for substantial usage of caching mechanisms, which is primarily used by the South Side Modules. For each CPE that connects to one of these Modules, very little data is actually read from the database, as much of what is needed is already cached locally in memory in each instance of the Module in question.

8.7 Modules

The Fusion is a highly modularized system. This ensures customer flexibility in choices such as preferred platforms (for the individual Modules), network topology for scalability and security concerns, minimal deployments according to needed functionality etc. The below sections provide a very brief introduction to the individual Modules, with references to further reading.

8.7.1 Fusion Database

The database Module contains the SQL schema and some basic installation documentation for the Fusion. The customer is responsible for being the database administrator.

Currently the Fusion can run on one database: MySQL. This database engine have performed very well, without a lot of tweaking. Even so, one should expect that some amount of tweaking is required to get a high performance on Fusion. Read more about Fusion Database in [6].

8.7.2 Fusion Web

The web interface is targeted at both customer support and system management. The GUI offers a broad range of operations to perform on the Fusion, such as:

- Search for Units
- Work with Units, Profiles and Unit Types
- Work with Groups and Jobs
- Search and view syslog messages
- Software management
- Inspection Mode provisioning
- Wizards to help with common operations

The web interface is mostly suited for single operations and one-on-one provisioning (Inspection Mode) with CPEs. It's also by far the most user-friendly management interface to the Fusion. Read more about Fusion Web in [5].

8.7.3 Fusion Shell

Fusion Shell is the command line interface to the Fusion. This Module is targeted at system management, especially operations that are repeated. Scripts can be created to automate large and/or complex tasks. Basically Fusion Shell offers the same functionality as Fusion Web, but in a less user-friendly and more powerful way. Some functionality is only available in Fusion Shell:

- Export the Fusion database or a Unit Type to files
- Import the Fusion database or a Unit Type from files
- Delete the Fusion database or a Unit Type in the database
- Remove unused system Parameters
- Add or change Unit specific Job Parameters
- Process so-called "Taiwan files" for factory provided CPE bootstrap information and configuration settings.
- Make complex scripts with if/else/loops logic

Read more about Fusion Shell in [4].

8.7.4 Fusion Web Services

The Web Services interface is meant for machine-to-machine communication, in other words for integration of any third-party system with the Fusion. For most larger Providers, provisioning is only a small part of what is needed to provide the end-user

with a service. When an end-user is created in the Providers systems, it must commonly be communicated to the Fusion. The most important functions are

- Add unit
- Change unit
- Move unit
- Delete unit
- Search units
- Retrieve unit

This interface is very limited as of now, but so far it has not been necessary with more functionality, as this interface will suit most basic purposes. This interface will be extended according to customer requirements in the future. Read more about Fusion Web Services in [7].

8.7.5 Fusion Core Server

The Fusion Core Server is required to use the Jobs functionality in the Fusion. The Fusion Core Server monitors the Jobs status and starts/stops the Jobs based on the stop rules. This Module has been developed to be able to deal with large numbers of Units without a negative impact on the performance on the Fusion database. It also handles report generation and syslog trimming. Read more about Fusion Core Server in [8].

8.7.6 Fusion Monitor Server

The Fusion Monitor Server monitors the availability of the other Fusion Modules, providing easy to understand status and history information presented by the Fusion Web module. The Fusion Monitor Server will store historic availability information such as event timestamps, uptime and monitoring messages for the Fusion Modules in the Fusion database. Read more about Fusion Monitor Server in [9].

8.7.7 Fusion Syslog Server

The Fusion Syslog Server is very useful if and only if you have devices that send messages to a syslog-server. In those cases you can monitor the events on the devices directly in Fusion Web.

8.7.8 Fusion TR-069 Server

The Fusion TR-069 Server is responsible for the communication between TR-069 enabled CPEs and the Fusion. This Module can be deployed on several hosts in order to scale up the performance in large deployments with TR-069 enabled devices. Read more about this server and test-client in [1].

8.7.9 Fusion SPP Server

The Fusion SPP Server is responsible for the communication between HTTP/HTTPS/TFTP/Telnet enabled devices and Fusion. The Module can be deployed on several hosts in order to scale up the performance in large deployments. Read more about this server in [2].