Service Suite Architecture 9.3

Add Tools

Added by Goran Jovanovic, last edited by Goran Jovanovic on Apr 10, 2014

This page documents the architecture of the 9.3 (Whitewater) release.

1. Introduction

This page provides a high level overview of the Service Suite Architecture. Each major topic is introduced on this page, and then refined in a child page. The child pages then provide further references to design and implementation pages.

For guidance on how to maintain these pages see Guidelines to Maintain Architecture Pages.

For a summary of the significant architectural changes since the last major release (r8) see Significant Changes from Previous Major Release below.

2. Architectural Goals

Architectural goals define what is important. The architectural goals for Service Suite are:

- **Scalability** - Ability to support all enterprises from small to large, with fifty to thousands of mobile workers and the corresponding dispatchers and work orders.
- **Openness** - Published interfaces for all functionality to allow enterprise applications to easily integrate with Service Suite.
- **Localization** - Internationalization to address local language and presentation differences.
- **Configurability** - No source changes required to support the work practices of individual customers. All configuration data is maintained in a database.
- **Highly available** - Workforce management is a mission-critical operational system. A multi-tier, distributed architecture with fail-over mechanisms is required.

Internal to Product Development we have the following additional goals in descending order of priority:

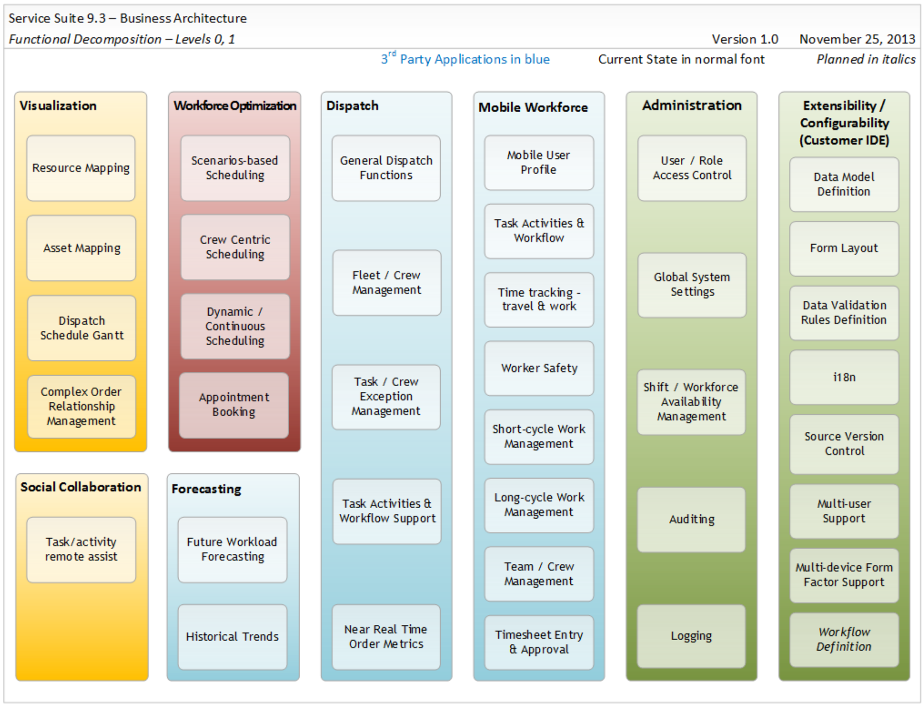
1. **Correctness** - It's got to work properly.
2. **Simplicity** - The simpler the code is, the easier it is to develop, test and maintain. Simplicity also makes achieving the other goals easier.
3. **Robustness** - Handle errors and unexpected conditions/inputs as gracefully as possible (while maintaining simplicity).
4. **Performance** - Responsiveness and throughput must be acceptable.

For more details see: Architectural Goals (Service Suite)

3. Business Architecture

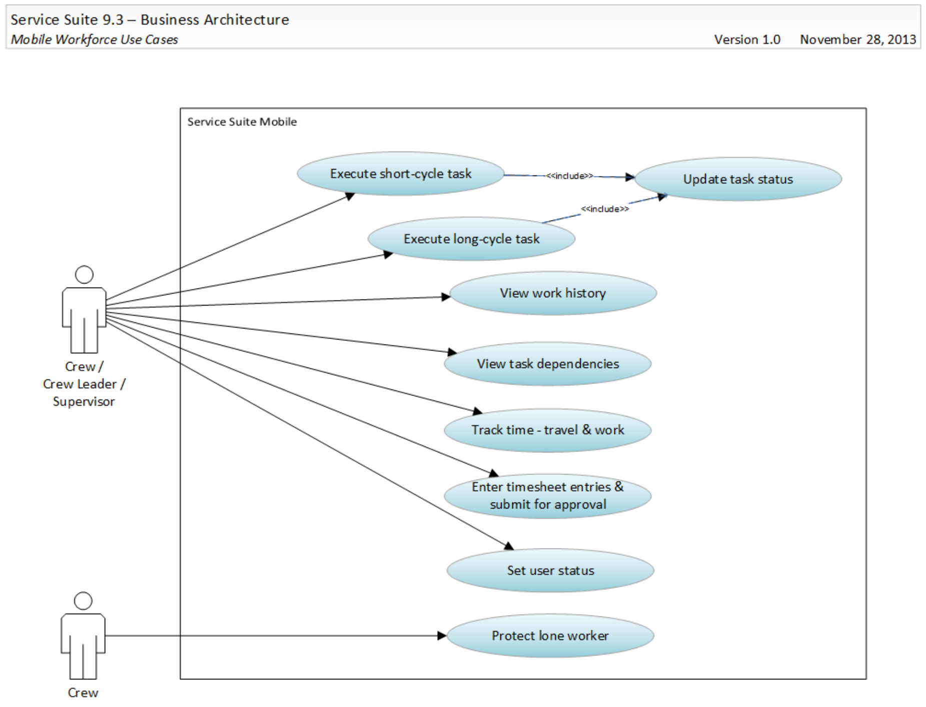
Business architecture adapted from TOGAF consists of high-level functional decomposition and use case diagrams.

3.1. Functional Decomposition Diagram

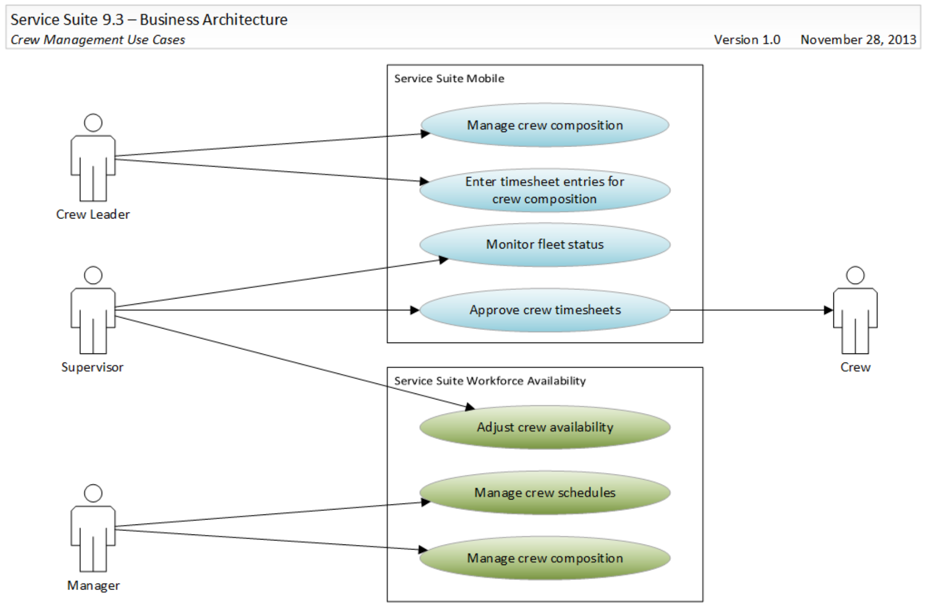

Service Suite 9.3 – Business Architecture
Functional Decomposition – Levels 0, 1
Version 1.0
November 25, 2013
3rd Party Applications in blue
Current State in normal font
Planned in italics
Visualization
Resource Mapping
Asset Mapping
Dispatch Schedule Gantt
Complex Order Relationship Management
Workforce Optimization
Scenarios-based Scheduling
Crew Centric Scheduling
Dynamic / Continuous Scheduling
Appointment Booking
Dispatch
General Dispatch Functions
Fleet / Crew Management
Task / Crew Exception Management
Task Activities & Workflow Support
Near Real Time Order Metrics
Mobile Workforce
Mobile User Profile
Task Activities & Workflow
Time tracking - travel & work
Worker Safety
Short-cycle Work Management
Long-cycle Work Management
Team / Crew Management
Timesheet Entry & Approval
Administration
User / Role Access Control
Global System Settings
Shift / Workforce Availability Management
Auditing
Logging
Extensibility / Configurability (Customer IDE)
Data Model Definition
Form Layout
Data Validation Rules Definition
i18n
Source Version Control
Multi-user Support
Multi-device Form Factor Support
Workflow Definition
Social Collaboration
Task/activity remote assist
Forecasting
Future Workload Forecasting
Historical Trends

3.2. Use Case Diagrams

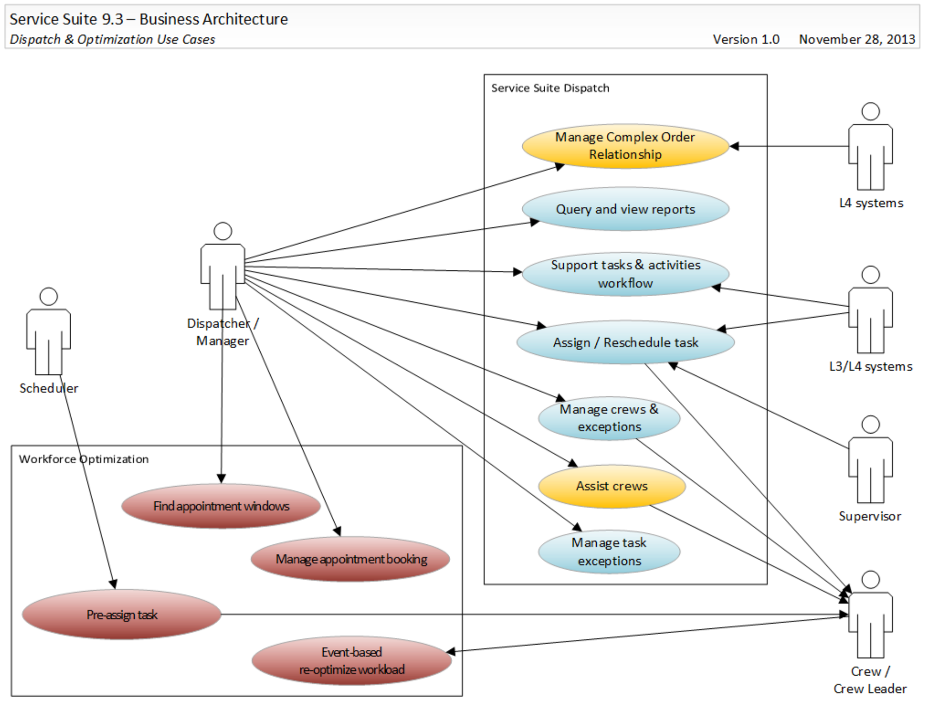
3.2.1. Mobile Workforce



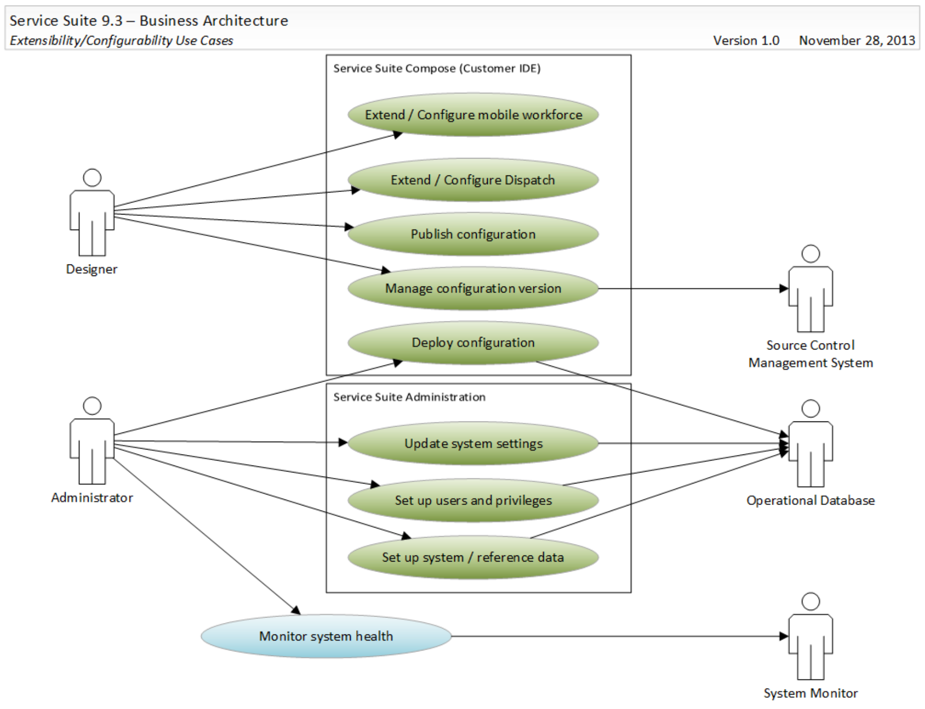
3.2.2. Crew Management



3.2.3. Dispatch and Optimization

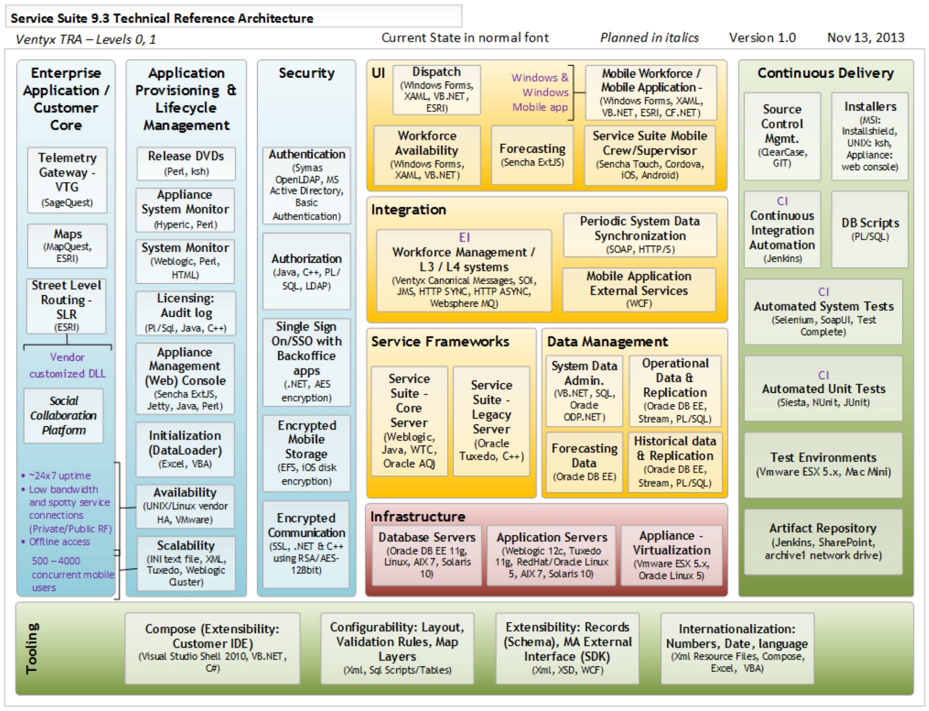

Service Suite 9.3 – Business Architecture
Dispatch & Optimization Use Cases
Version 1.0 November 28, 2013
Service Suite Dispatch
Manage Complex Order Relationship
Query and view reports
Support tasks & activities workflow
Assign / Reschedule task
Manage crews & exceptions
Assist crews
Manage task exceptions
L4 systems
L3/L4 systems
Supervisor
Crew / Crew Leader
Dispatcher / Manager
Scheduler
Workforce Optimization
Find appointment windows
Manage appointment booking
Pre-assign task
Event-based re-optimize workload

3.2.4. Extensibility and Configurability


Service Suite 9.3 – Business Architecture
Extensibility/Configurability Use Cases
Version 1.0 November 28, 2013
Service Suite Compose (Customer IDE)
Extend / Configure mobile workforce
Extend / Configure Dispatch
Publish configuration
Manage configuration version
Deploy configuration
Designer
Source Control Management System
Service Suite Administration
Update system settings
Set up users and privileges
Set up system / reference data
Administrator
Operational Database
Monitor system health
System Monitor

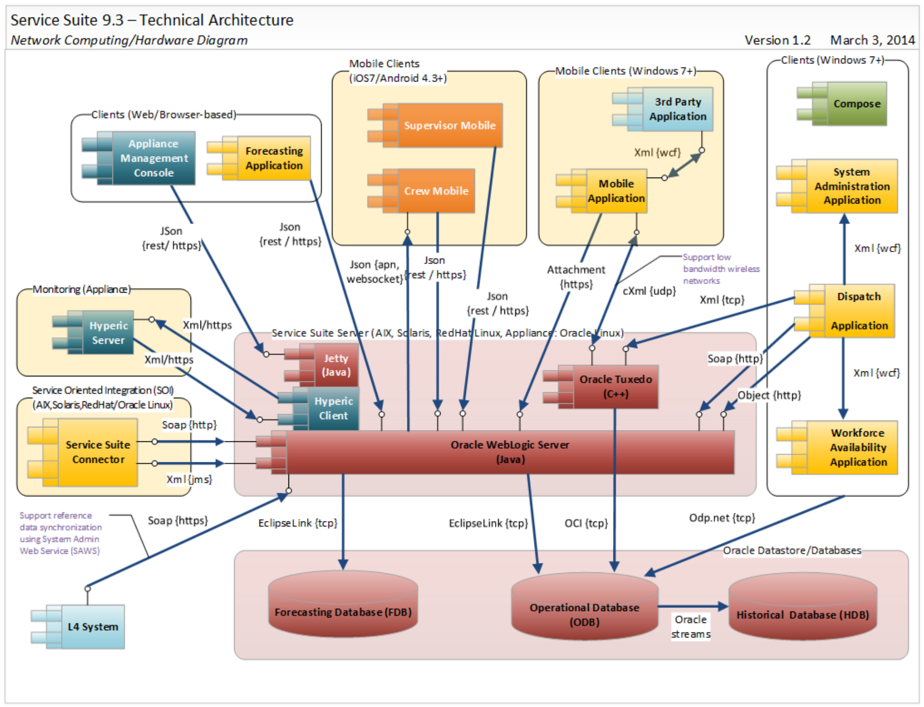
4. Technical Reference Architecture

Technology architecture is adapted from TOGAF Technical Reference Model.


Service Suite 9.3 Technical Reference Architecture
Ventyx TRA – Levels 0, 1
Current State in normal font
Planned in italics
Version 1.0
Nov 13, 2013
Enterprise Application / Customer Core
Telemetry Gateway - VTG (SageQuest)
Maps (MapQuest, ESRI)
Street Level Routing - SLR (ESRI)
Vendor customized DLL
Social Collaboration Platform
~24x7 uptime
Low bandwidth and spotty service connections (Private/Public RF)
Offline access
500 – 4000 concurrent mobile users
Application Provisioning & Lifecycle Management
Release DVDs (Perl, ksh)
Appliance System Monitor (Hyperic, Perl)
System Monitor (Weblogic, Perl, HTML)
Licensing: Audit log (Pl/Sql, Java, C++)
Appliance Management (Web) Console (Sencha ExtJS, Jetty, Java, Perl)
Initialization (DataLoader) (Excel, VBA)
Availability (UNIX/Linux vendor HA, VMware)
Scalability (INI text file, XML, Tuxedo, Weblogic Cluster)
Security
Authentication (Symas OpenLDAP, MS Active Directory, Basic Authentication)
Authorization (Java, C++, PL/SQL, LDAP)
Single Sign On/SSO with Backoffice apps (.NET, AES encryption)
Encrypted Mobile Storage (EFS, iOS disk encryption)
Encrypted Communication (SSL, .NET & C++ using RSA/AES-128bit)
UI
Dispatch (Windows Forms, XAML, VB.NET, ESRI)
Windows & Windows Mobile app
Mobile Workforce / Mobile Application - (Windows Forms, XAML, VB.NET, ESRI, CF.NET)
Workforce Availability (Windows Forms, XAML, VB.NET)
Forecasting (Sencha ExtJS)
Service Suite Mobile Crew/Supervisor (Sencha Touch, Cordova, iOS, Android)
Integration
EI
Workforce Management / L3 / L4 systems (Ventyx Canonical Messages, SOI, JMS, HTTP SYNC, HTTP ASYNC, Websphere MQ)
Periodic System Data Synchronization (SOAP, HTTP/S)
Mobile Application External Services (WCF)
Service Frameworks
Service Suite Core Server (Weblogic, Java, WTC, Oracle AQ)
Service Suite - Legacy Server (Oracle Tuxedo, C++)
Data Management
System Data Admin. (VB.NET, SQL, Oracle ODP.NET)
Operational Data & Replication (Oracle DB EE, Stream, PL/SQL)
Forecasting Data (Oracle DB EE)
Historical data & Replication (Oracle DB EE, Stream, PL/SQL)
Infrastructure
Database Servers (Oracle DB EE 11g, Linux, AIX 7, Solaris 10)
Application Servers (Weblogic 12c, Tuxedo 11g, RedHat/Oracle Linux 5, AIX 7, Solaris 10)
Appliance - Virtualization (Vmware ESX 5.x, Oracle Linux 5)
Continuous Delivery
Source Control Mgmt. (ClearCase, GIT)
Installers (MSI: Installshield, UNIX: ksh, Appliance: web console)
CI Continuous Integration Automation (Jenkins)
DB Scripts (PL/SQL)
CI Automated System Tests (Selenium, SoapUI, Test Complete)
CI Automated Unit Tests (Siesta, NUnit, JUnit)
Test Environments (Vmware ESX 5.x, Mac Mini)
Artifact Repository (Jenkins, SharePoint, archive1 network drive)
Tooling
Compose (Extensibility: Customer IDE) (Visual Studio Shell 2010, VB.NET, C#)
Configurability: Layout, Validation Rules, Map Layers (Xml, Sql Scripts/Tables)
Extensibility: Records (Schema), MA External Interface (SDK) (Xml, XSD, WCF)
Internationalization: Numbers, Date, language (Xml Resource Files, Compose, Excel, VBA)

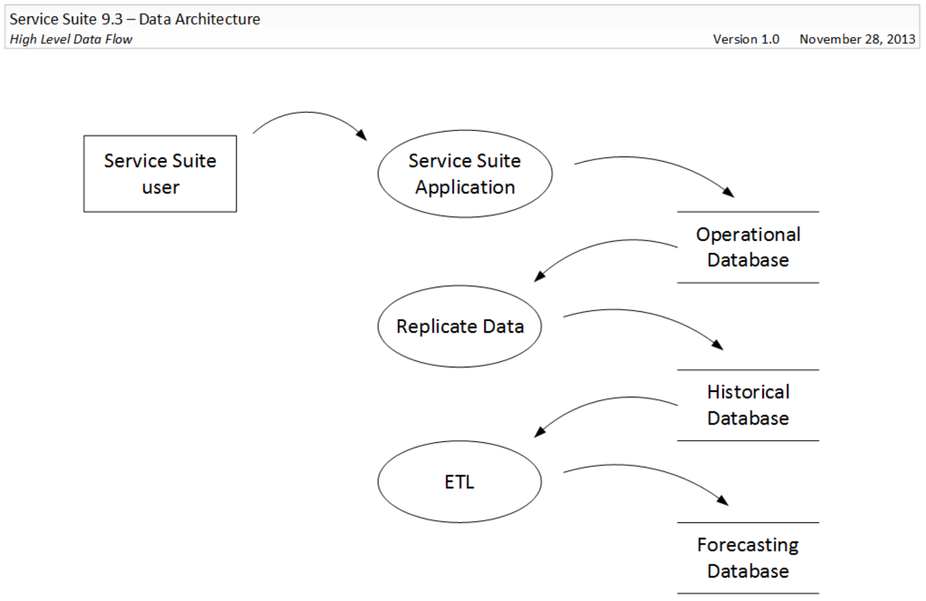
5. Networked Computing Diagram

The networked commuting diagram is adapted from TOGAF.


Service Suite 9.3 – Technical Architecture
Network Computing/Hardware Diagram
Version 1.2
March 3, 2014
Clients (Web/Browser-based)
Appliance Management Console
Forecasting Application
Mobile Clients (iOS7/Android 4.3+)
Supervisor Mobile
Crew Mobile
Mobile Clients (Windows 7+)
3rd Party Application
Xml {wcf}
Mobile Application
Clients (Windows 7+)
Compose
System Administration Application
Xml {wcf}
Dispatch Application
Xml {wcf}
Workforce Availability Application
JSon {rest/ https}
Json {rest / https}
Json {apn, websocket}
Json {rest / https}
Json {rest / https}
Attachment {https}
cXml {udp}
Support low bandwidth wireless networks
Xml {tcp}
Monitoring (Appliance)
Hyperic Server
Xml/https
Xml/https
Service Suite Server (AIX, Solaris, RedHat Linux, Appliance: Oracle Linux)
Jetty (Java)
Hyperic Client
Oracle Tuxedo (C++)
Soap {http}
Object {http}
Service Oriented Integration (SOI) (AIX,Solaris,RedHat/Oracle Linux)
Service Suite Connector
Soap {http}
Xml {jms}
Oracle WebLogic Server (Java)
Support reference data synchronization using System Admin Web Service (SAWS)
Soap {https}
EclipseLink {tcp}
EclipseLink {tcp}
OCI {tcp}
Odp.net {tcp}
Oracle Datastore/Databases
L4 System
Forecasting Database (FDB)
Operational Database (ODB)
Oracle streams
Historical Database (HDB)

6. Data Architecture

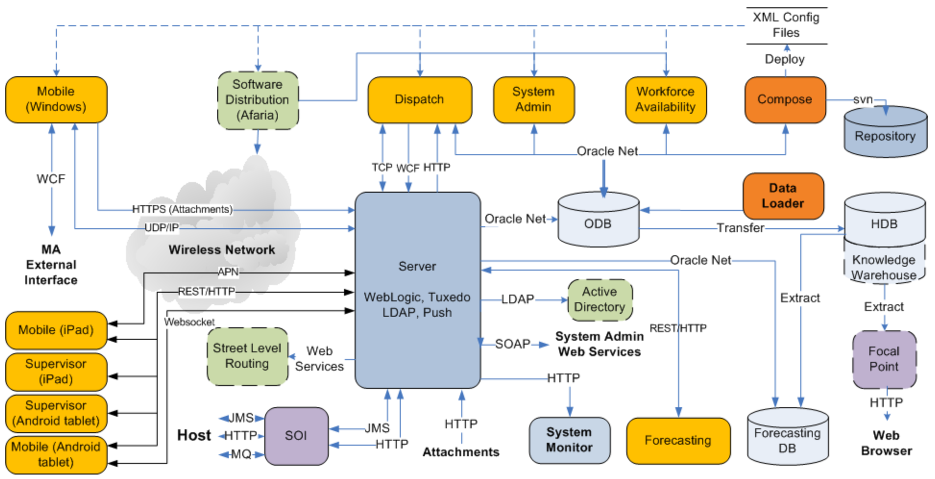
Data Flow is adapted from TOGAF.



7. System Context

Service Suite managed the work of mobile users (predominantly) in the utilities and telecom sectors. Service Suite supports short cycle work (eg. end user install and repair by appointment) and long cycle work (construction).

7.1. System Components



[Picture, Source]

- **Mobile (Windows) Client**: Is used to perform work in the field. The user can view assigned orders, report progress on orders, fill in forms, and communicate with other users or host systems. Mobile Mapping (optional feature) provides a geographical view of the user's orders and territory. The MA Client generally runs on a laptop/notebook PC or Windows Mobile device.
- **Mobile (iPad/Android tablet) Client**: A version of the mobile client that runs on Apple iPad and Android tablet devices.
- **Supervisor (iPad/Android tablet) Client**: A supervisor mobile client that runs on Apple iPad and Android tablet devices. The Supervisor client is implemented as a separate application for iPad and Android tablet devices.
- **Dispatch (DA) Client**: Is used to dispatch work and monitor the progress of work in the field. The focus is on providing tools for exception handling, allowing dispatchers to take proactive action in meeting service level agreements. Dispatch Mapping (optional feature) provides a geographical view of the dispatcher's domain. The DA Client generally runs on a Desktop PC.
- **Workforce Availabiliy (WA) Client**: Is used to manage mobile user's work schedules. The WA Client generally runs on a Desktop PC or Laptop PC.

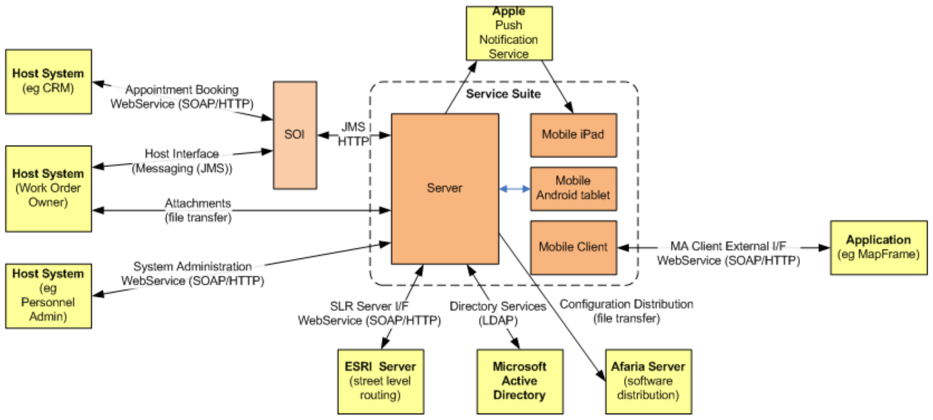
- **System Administration (SA) Client**: Is used manage system data, such as users, areas, job codes, etc. It includes the PickList Editor (PE). The SA Client generally runs on a Desktop PC.
- **Compose Client**: Compose provides the ability to configure the Service Suite application, such as defining records, forms and messages. The configuration data is stored in a Subversion CM system. Multiple users can edit the same configuration using the Compose Client. When configration changes are finalized, Compose updates the server's configuration. Clients need to be updated using a third-party distribution mechanism.
- **Dataloader:** Dataloader is used to load system data, such as users, areas, order codes, job codes, etc. in a bulk operation at the beginning of a project or during migration for an upgrade. The Dataloader is an Excel spreadsheet application.
- **Host Systems**: The "Host" represents the upstream owner of work orders. It can be a single system or many systems. In the latter case it is the middle-ware's responsibility to route upstream messages to the right system. Host systems created work orders, and are notified of progress on the work orders.
- **SOI**: Ventyx Service Oriented Integration Platform provides a holistic way to integrate among software products within Ventyx and ABB, and with third party systems. Canonical messages are used to convey business processes among the applications. Service Suite host interface and AB web services are encapsulated and transformed into canonical messages equivalents.
- **Street Level Routing (SLR)**: Some custom software from ESRI that runs on top of the ESRI ArcGIS engine and Network Analysts component to provide the most optimal route for a day's work for mobile user based on GIS information.
- **Service Suite Server**: Is the processing component of Service Suite. It runs on a Unix/Linux server, and, for demonstration purposes in 9.2.0, in a VMware virtual environment.
- **Operational Database (ODB)**: The power house to store all operational and system data for Service Suite.
- **Historical Database (HDB)**: HDB provides historical and auditing information about the ODB. Near-real-time information can be extracted out for KPI, knowledge warehouse, business intelligence and forecasting purposes.
- **User Directory (LDAP)**: A directory service that stores user authentication related information. It can either be a built-in LDAP server or Microsoft Active Directory.
- **System Monitor**: is used to monitor the health of Service Suite and provides system alerts.
- **Software Distribution**: is used to distribute configuration updates for MA notebook edition, MA Windows Mobile edition, and the back office applications DA, SA, and WA. In addition, it can be used to install updated version of MA notebook edition and retrieve log files from MA notebook edition.
- **Knowledge Warehouse**: provides audit and historical information on the operation of Service Suite so that data-mining applications can analyse the data from it.
- **Forecasting Client**: is used as analytical tool to help predict future work demand based on historical trends in customer data. The forecasting client is web application and run in an internet browser.
- **Forecasting Database:** Stores forecasting data extracted from the HDB and imported from external sources (weather, ...).

7.2. System Users (Roles)

- **Technician**: Perform work in the field. Technicians notify the system when they begin traveling to and arrive on site at a service request, and transmit job completion reports when they complete the service request. Technicians can have timesheet records and submit them for approval. Technicians use the MA client.
- **Crew Leader**: A technicians and/or supervisors that represents multiple mobile users working as a team. The composition of the crew may be pre-defined or dynamically etsablished and changed during the shift.
- **Supervisor**: A mobile user that supervises a group of technicians and/or crews. Supervisors can perform all of the functions of technicians. Supervisors also have access to the status of each supervised technican/crew, a summary of the orders assigned to their technicans/crews, and the details of each order assigned to their technicans/crews. Supervisors can manage the assignment of their technicans/crews to shifts, and also approve timesheets of their technicans/crews. Supervisors use the MA and WA clients, and may also be provided with limited access to the DA client.
- **Call-taker**: Receives service order requests from customers and enter the requests directly into Service Suite. Service Suite provides call-takers using the system with the status of each orderand each mobile user, and the details of each order. Call-takers use the DA client.
- **Dispatcher**: Can perform all of the functions of call-takers. Dispatchers also have the capability to reschedule orders, manage the assignment of orders to mobile users, perform order progression for mobile users, and manage the assignment of mobile users to shifts. Dispatchers use the DA client.

- **Manager**: Can perform all of the functions of dispatchers and system administrators. Managers also perform special functions with the mapping application. Managers use the DA, WA, SA and PE clients.
- **System Administrator**: Maintains the system tables and parameters (such as job codes and skills) and has the capability to assign mobile users to shifts. System Administrators use the SA, WA and PE clients, and may also use the Compose client.
- **System Designer**: Makes changes to the configuration (records, forms, messages). System Designers use the Compose Client.

8. External Interfaces


Apple Push Notification Service
Host System (eg CRM)
Appointment Booking WebService (SOAP/HTTP)
Service Suite
SOI
JMS | HTTP
Mobile iPad
Host Interface (Messaging (JMS))
Host System (Work Order Owner)
Server
Mobile Android tablet
Attachments (file transfer)
Mobile Client
MA Client External I/F WebService (SOAP/HTTP)
Application (eg MapFrame)
System Administration WebService (SOAP/HTTP)
Host System (eg Personnel Admin)
SLR Server I/F WebService (SOAP/HTTP)
Directory Services (LDAP)
Configuration Distribution (file transfer)
ESRI Server (street level routing)
Microsoft Active Directory
Afaria Server (software distribution)

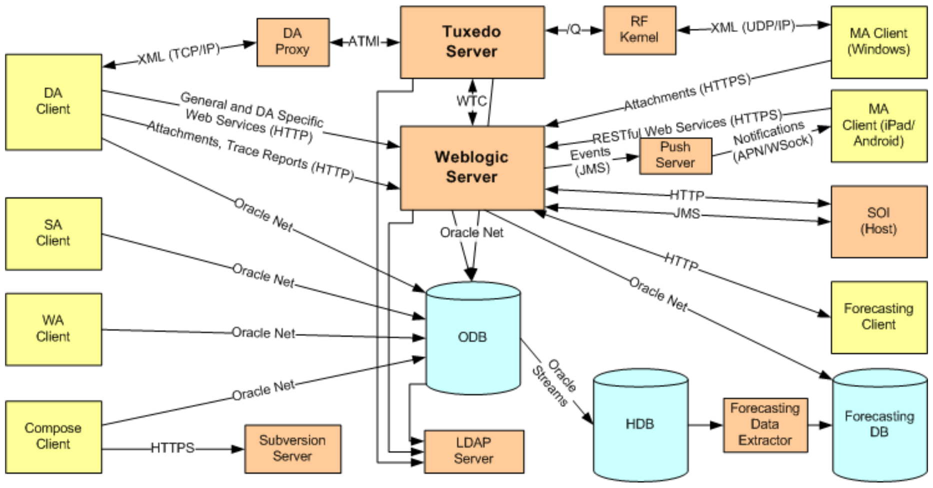
[Picture,Source]

Service Suite provides the following external interfaces:

- **Canonical Messages via SOI**: This is the main interface to upstream systems. It is a messaging interface using JMS and/or HTTP. The host system sends requests to manage work orders. Service Suite sends notifications of work order and mobile user status changes.
- **Appointment Boooking Web Service**: The ABWS is used to find appointment windows and book appointments.
- **System Administration Web Service**: The SAWS is used to manage users (office and mobile) and to manage mobile user's work schedules (shifts).
- **SLR Server Interface**: Service Suite can be integrated with ESRI's SLR server to schedule work orders based on street level routing.
- **MA Client External Interface**: This interface can be used to integrate other applications into the mobile clients. Its main use cases are to retrieve update form input (eg automatic meter reads) or to send details form data (eg to a GIS/map software).
- **Apple Push**: Apple's push service is used to send notifications to the iOS version of the MA Handheld Client.
- **Directory Services Interface**: The LDAP interface allows Service Suite to use an external directory to authenticate its users.
- **Software Distribution Interface**: Service Suite can be integrated with Afaria Software Distribution to distribute configuration data to the client devices.

For more details see: External Interfaces (Service Suite)

9. Major Internal Interfaces



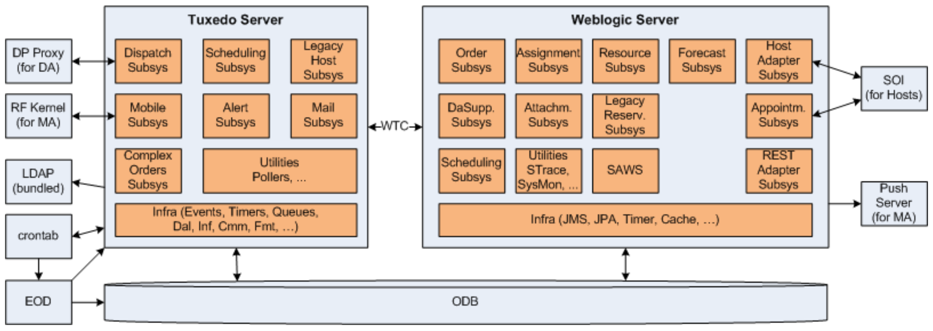
[Picture, Source]

Service Suite consists of the server, the databases and the clients. The server has a Tuxedo (C++) and Weblogic (Java) part. The main interfaces between these components are:

- **DA Client**:
  - XML: This is the legacy protocol between the DA client and the Tuxedo server. It is used mainly for displays (viewports).
  - Web Services: The DA client uses multiple web services to manage orders, mobile users, etc: AppointmentWebService, OrderLifecycleWSService, OrderProgressionWSService, AssignmentWSService, SchedulingWebService, DAResourceWSService, HostWSService (misc. operations).
  - Attachments: The DA clients uses the Attachment Web server to download and upload attachments.
  - Oracle Net: The DA client accesses the ODB directly for reports and details display when hovering on the Gantt Chart in Dispatch Schedule.
- **SA Client**:
  - Oracle Net: The SA client is a 2-tier client. It accesses the ODB directly.
- **WA Client**:
  - Oracle Net: The WA client is a 2-tier client. It accesses the ODB directly.
- **MA Client** (Notebook and Windows Mobile editions):
  - XML: This is the legacy protocol using a proprietary way to encode binary XML similar to ASN.1. It is used for virtually all interactions between MA and server.
  - Attachments: The MA client uses the Attachment Web server to download and upload attachments.
- **MA Client (iPad)**:
  - RESTful Web Services: The edition of the MA Client for the iOS. It uses RESTful style Web Services over HTTPS.
  - Apple Push Notification (APN): Used for server to client notifications.
- **MA Client (Android tablet)**:
  - RESTful Web Services: The edition of the MA Client for other devices. It uses RESTful style Web Services over HTTPS.
  - WebSocket: Bi-directional, persistent TCP connection based on HTML5 standard. Currently used for server to client notifications.
- **Compose Client**:
  - HTTPS: The Compose client uses secure HTTP to access the configuration repository on the Subversion server.
  - Oracle Net: The Compose client accesses the ODB to deploy configuration data.
- **Server**: Besides the interfaces mentioned above, the server uses:
  - WTC: For interactions between Tuxedo and Weblogic. This is a ATMI interface using the Weblogic Tuxedo Connector (WTC).
  - Oracle Net: To access the ODB.
- **Databases**: Besides the interfaces mentioned above, the databases use:
  - Streams: To apply changes in the ODB to the HDB.
  - Forecasting Extractor: To extract forecasting data from HDB and import to Forecasting DB.
- **Forecasting Client:**
  - RESTful Web Services.

For more details see: Internal Interfaces (Service Suite)

10. Server Architecture



[Picture, Source]

The Service Suite Server consist of two main parts:

- **Tuxedo Server**: This is the legacy Tuxedo domain, using C++. It still provides the following major functionality: DA session, viewports, MA session, alerts, scheduling, complex orders, text messaging, etc.
- **Weblogic Server**: The WLS domain provides all functionality implemented in Java, Web Services and Web Servers (HTTP). Specifically it provides the following functionality: Order processing, host interface, etc.

In addition to the two major components, the server also includes:

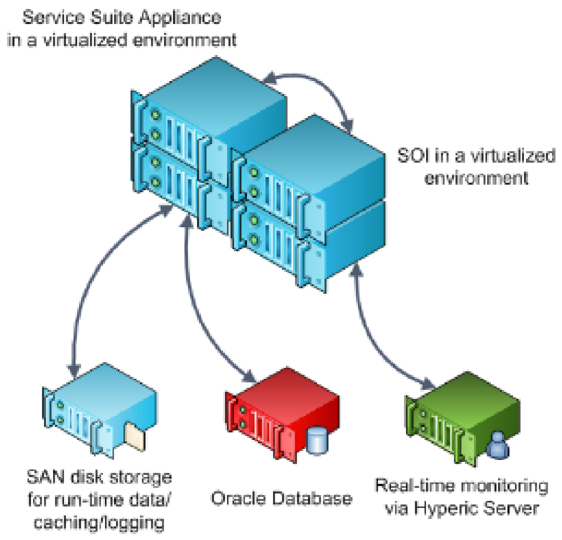
- **DP Proxy**: This is a Tuxedo client that provides the access point for the DA Clients. It mediates between the TCP/IP based protocol of the DA Client and the ATMI interfaces in the Tuxedo Domain.
- **RF Kernel**: The RF Kernel is the access point for mobile clients (classic). It consists of a set of plain Unix/Linux and Tuxedo client processes. It mediates between the UDP/IP based protocol of the MA Client and the ATMI interfaces in the Tuxedo domain.
- **SOI**: The SOI connector integrates the server with other Ventyx systems and host systems. It adapts Service Suite to the canonical messages.
- **LDAP**: The server includes an embedded LDAP server. It is used when there is no external directory server to authenticate Service Suite users with.
- **crontab**: The Unix/Linux crontab feature is used to initiate periodically scheduled activities and batch jobs, such as WO runs, EOD processing, etc.
- **Push Server**: A stand-alone server that pushes notifications to MA clients when there is a change to their assignment.

Also see Major Internal Interfaces above.

For more details see: Server Architecture (Service Suite)

11. Service Suite Appliance

To reduce the Total Cost of Ownership (TCO) and Total Cost of Development (TCD) of Service Suite, an appliance solution is proposed. A software appliance is a pre-integrated and ready to run image of Service Suite server with all the required 3rd party components and a stripped down version of an operating system. The appliance is supported under VMware vSphere environment. The following shows a conceptual environment where Service Suite appliance is to be run:


Service Suite Appliance in a virtualized environment
SOI in a virtualized environment
SAN disk storage for run-time data/ caching/logging
Oracle Database
Real-time monitoring via Hyperic Server

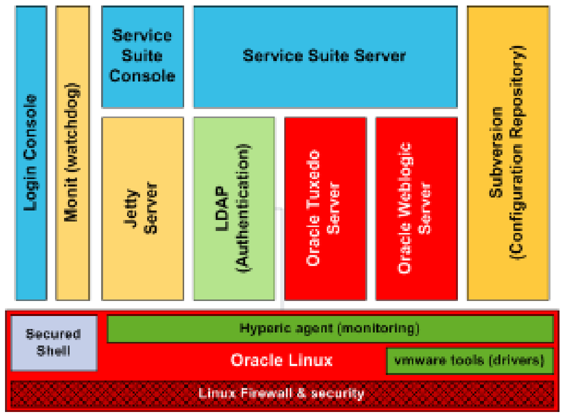
The key concept is that the Service Suite appliance, the SOI and Hyperic Server will exist as virtual machines in a virtualized environment. If an SLR server is required, it will also exist in yet another virtual machine in the virtualized environment. To ease the upgrade process, an appliance is completely replaced by a new instance during an upgrade so the software image in an appliance is always consistent and known. The run-time data, data caches, and logs from an existing instance are stored outside the appliance in a virtual disk that can be moved from one virtual machine to a different one during the upgrade process.

Mentioned Hyperic Server will be used to monitor the virtual environment, Service Suite appliance, the SOI core, and/or the Oracle database in real-time, and alert the administrator in a monitoring console or via email of any potential problems.

For details about VM hardware requirements see: Service Suite Appliance Sizing

11.1. Key Components of the Server Appliance

The key components within a Service Suite appliance are basically the same as required by a regular Service Suite server. The following diagram shows the key components:



Because an appliance is a closed system, i.e., a customer has no write access to the operating system or any software inside, extra management capabilities are required to support a day-to-day operation of Service Suite in an enterprise environment. Security is a key component in the appliance. The Linux firewall is turned on always and only a couple of ports required by Service Suite are open. No other standard ports are open.

The following is a brief description of each key component:

Oracle Linux: This is equivalent to the RedHat Enterprise Linux.

Linux Firewall & security: The Linux firewall to accept only certain traffic and other built-in Linux security components.

Secured Shell: A support account can be enabled to allow access to the Linux environment in case something goes really wrong. Only Ventyx staff will be given access to this console.

vmware tools: Enhanced drivers from VMware to run the virtual machine smoothly in a VMware vSphere environment. It also allows the virtual machine to shutdown gracefully if the virtual machine is shut down externally from VMware management console.

Hyperic agent: This agent is responsible for collecting operating system statistics and operating data from key components and Service Suite. The collected data are sent to the Hyperic server and stored there for further processing and alert generation.

Jetty Server: A light-weighted Java web server to host the Service Suite management console. It is started when the appliance is booted.

Service Suite console: The web-based management console of Service Suite. It replaces install.ksh, BldRunTime, StartServer, ShutdownServer, etc. from a normal Service Suite server with an intuitive web-based user interface with some basic monitoring capabilities.

LDAP: The same LDAP server that is bundled with Service Suite. However, the LDAP server is started as soon as the virtual machine is started in the appliance.

Oracle Tuxedo Server: The ATMI core to host C++ code for Service Suite. See the Server Architecture sections for more information.

Oracle Weblogic Server: The Java EE container to host Java code for Service Suite. See the Server Architecture sections for more information.

Service Suite Server: The same Service Suite server system as the normal one. It has a medium size pre-configured and only one size is supported. The strategy to support a large size system will be revealed later. See the Server Architecture sections for more information.

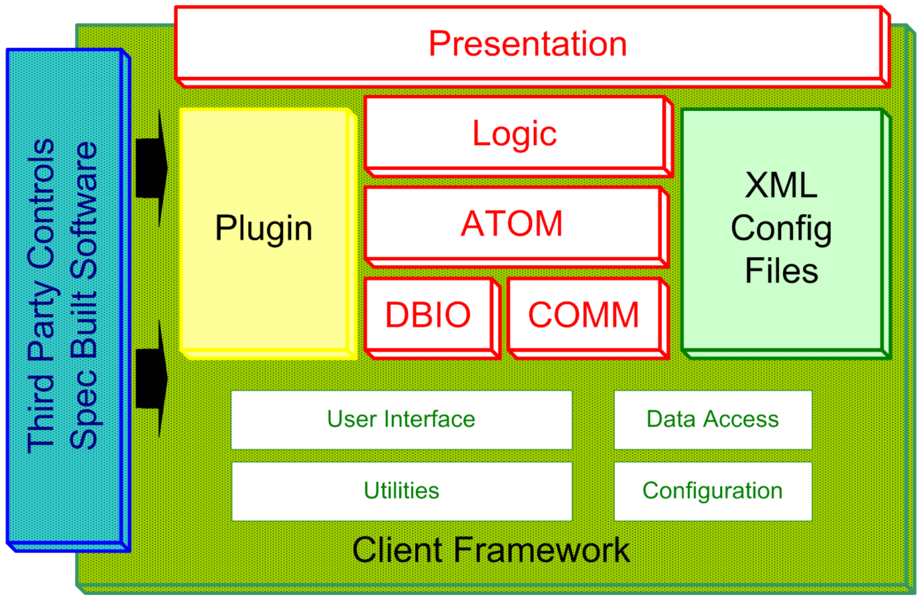
Subversion Configuration Repository: The same subversion server that is used to store Compose configuration and its project. See the Compose Architecture sections for more information.

Login Console: The text based console that replaces login prompt on the VMware console window. Can be used to enable/disable support account, configure network settings and to shutdown the box.

Monit: A watchdog service used to make sure Jetty, LDAP and Subversion services are healthy and running.

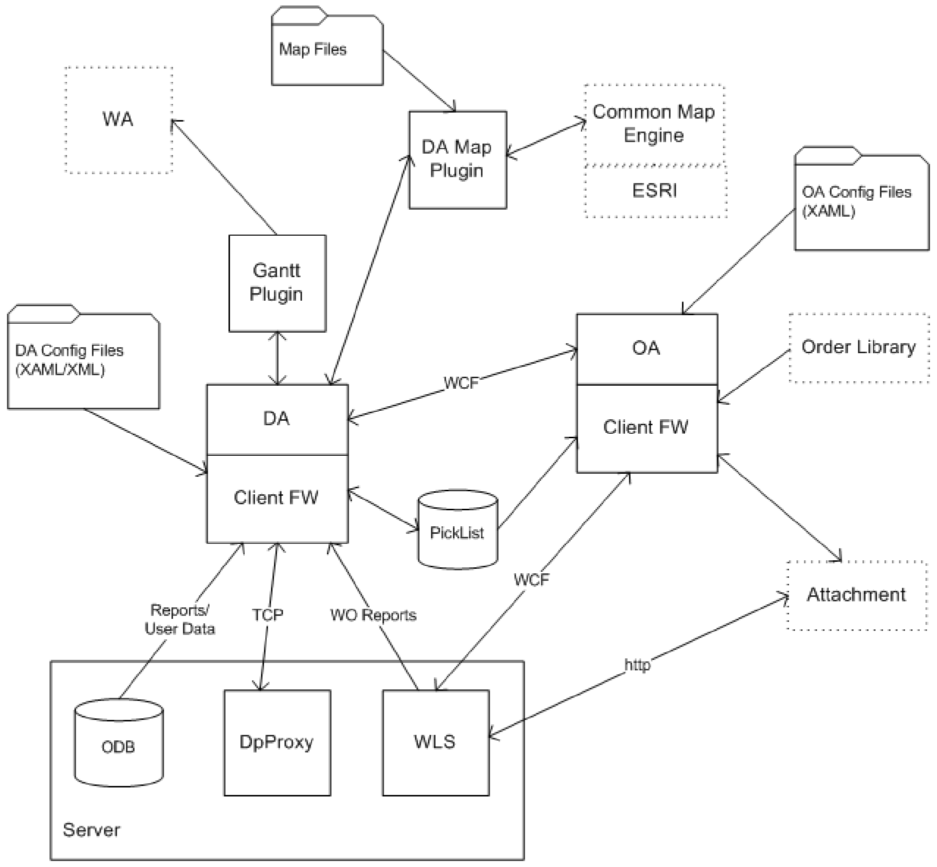
12. Client Architecture

12.1. Client Framework



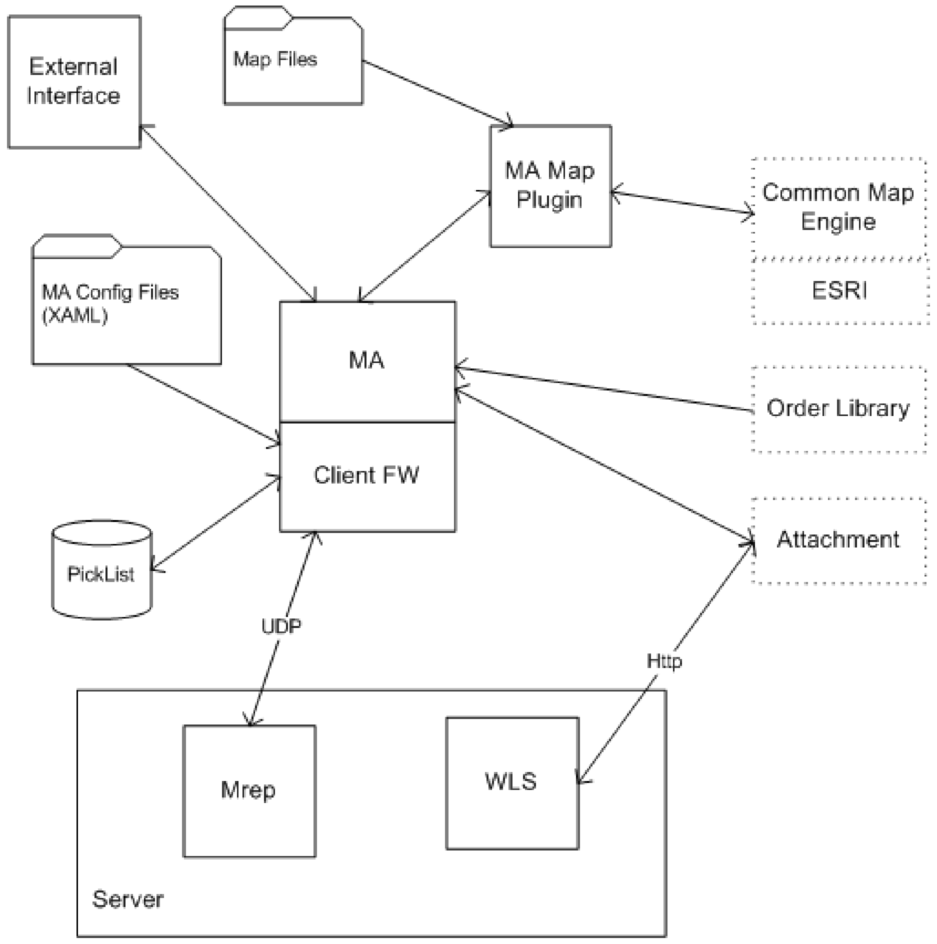
For more details see: [Service Suite 8.0 Client Framework]

12.2. DA



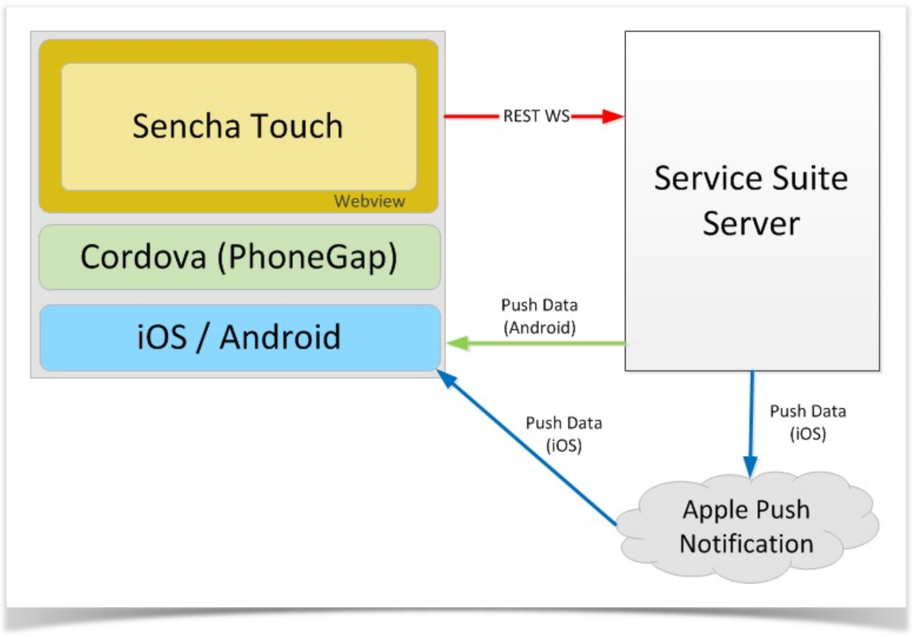
[Picture]
For more details see: DA Architecture

12.3. MA (Notebook Edition)



[Picture]
For more details see: MA Architecture

12.4. Mobile and Supervisor (iPad and Android Tablet)

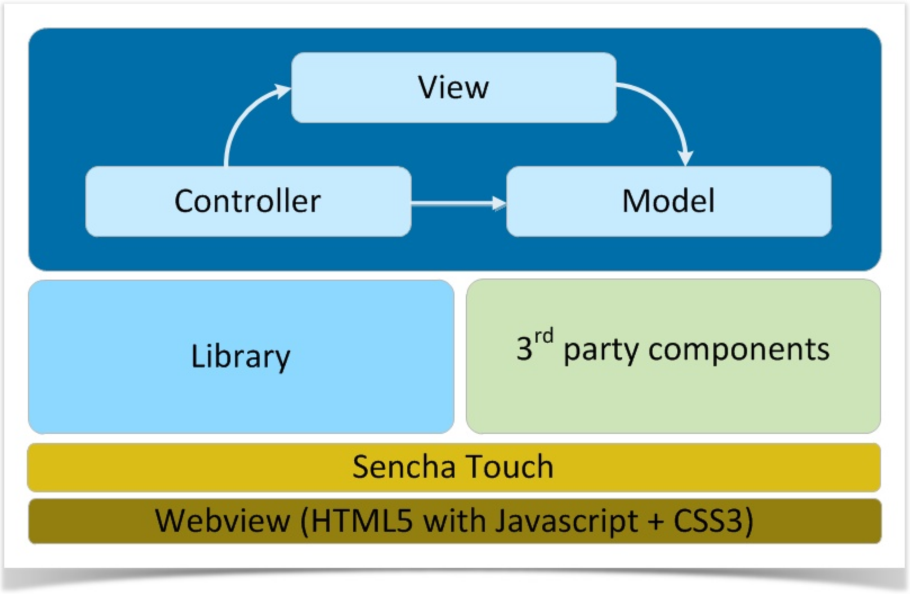


The mobile is mostly designed and implemented using Sencha Touch 2.2.1 HTML5 framework. It is wrapped inside Cordova to make it like a native application. The current release runs on iPad using iOS 7+ and any Android tablets using Android 4.3+. Plugins, which are adapters between the Webview Javascript code and native code, are used primarily to provide functions only made possible by using native code. The same Javascript interface is used whether the underlying operating system is iOS or Android. There are 3 plugins implemented:

- Push notifications
- Navigation using native maps
- Email Composer for sending diagnostic log files

The following are some key architecture highlights in the mobile application:

- Using Cordova (PhoneGap) as the native application wrapper, it has a webview that holds 99% of the mobile application except the plugins as noted above.
- The mobile application is, in essence, a mobile web application which can also be run in a web browser (but the full official support for web browser will be available in a future release.)
- Supervisor app on iPad and Android tablet shares the same architecture as the mobile app except that there is no push implemented.
- Single-Sign-On is implemented between the Mobile and Supervisor apps using URL scheme by passing user ID and password from Mobile to Supervisor app.
- Communications, except push notifications, are done purely in webview using Restful web services (JSON).
- A bootstrap URL is required to pass in the host/IP address, port, and logical name to the native application so that it knows where the Service Suite server is. The logical name can be any text to identify the server.
- The mobile web application within the webview is downloaded from Service Suite server when the native application is run and is cached on the device using HTML5 application cache.
- If a newer version is available, it will automatically download a newer version into the device.
- WebSQL and HTML5 localstorage are used to cache data to support occasional off-line scenarios.
- Indexed DB SHIM (simulator) is used to store the log files as it is a NoSQL database. Although Indexed DB is an HTML5 feature, it is not available in iOS 7 and partially implemented in Android. The SHIM uses WebSQL as an underlying database.
- Application configuration is downloaded automatically using a REST web service call upon successful sign in. It is cached until a newer version is available.
- Pick list is downloaded automatically using HTML5 application cache mechanism.
- Push notifications on iOS devices are done using Apple Push Notification technology.
- Push notifications on Android devices are done using websocket implemented at the native layer (This will be changed to use Google Cloud Messaging in the future.)
- The mobile web application is implemented using Sencha Touch framework which has a Model-View-Controller (MVC) pattern. See the diagram below.



Component	Description
View	Contain all static layouts and placeholders for Compose configured forms
Controller	Contain all presentation logics and calls business logic to tie the application together
Model	Contain schema to represent data and metadata used in the application
Library	Contain business logics, Ventyx UX library hooks (styling), components like repeating group, multi-level pick list, form builder, pick list builder, off-line queue, validation engine hooks, etc.

3rd party components	Contain slide navigator, date-time and time controls, leaflet Javascript library + plugins, and work flow engine
Sencha Touch	The complete Sencha Touch + Charts SDK

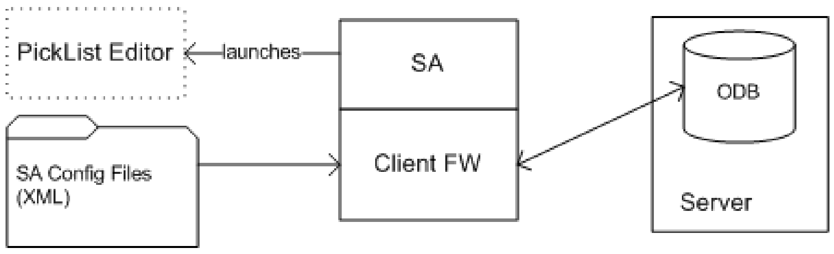
The application has the following structure:

Folder	Description
platforms/ios	Cordova iOS native code + plugin native code portion
platforms/android	Cordova Android native code + plugin native code portion
www-src/app	Mobile web application
www-src/resources	All resources (External Javascript libraries, SASS, images, generated CSS)
www-src/vmux	Ventyx UX framework like structure for common styling and fonts
www-src/touch	Sencha Touch source code and library
www-src/.sencha	Sencha Command files for Sencha Touch project
www-src/Test	Play framework source code + Siesta unit tests for running automated unit tests

The following tools are used for Sencha Touch mobile web application development:

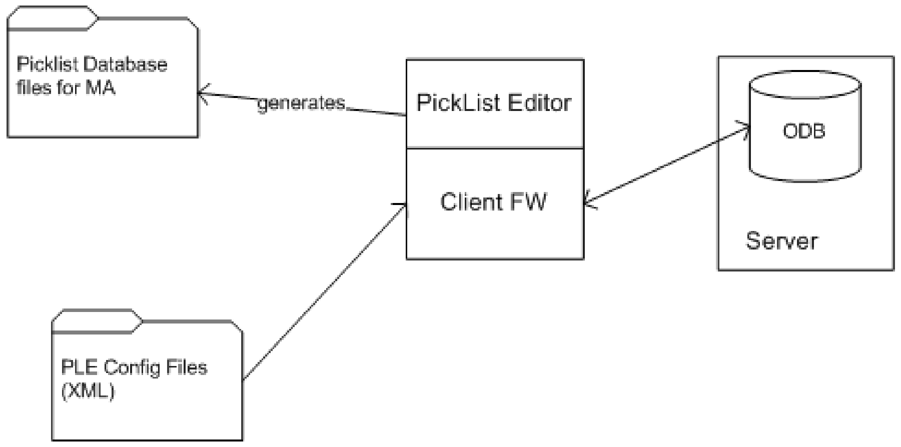
- Webstorm (from JetBrains)
- Sencha Touch + Charts v2.2.1 SDK (Commercial licence)
- Sencha Command v3.1.x
- Siesta unit testing framework
- GIT and SourceTree (A UI for GIT) are used for source control
- MacBook Pro

12.5. SA



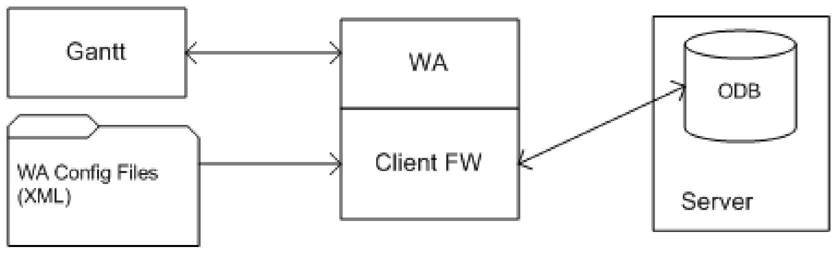
[Picture]
For more details see: SA Architecture

12.6. PickList Editor



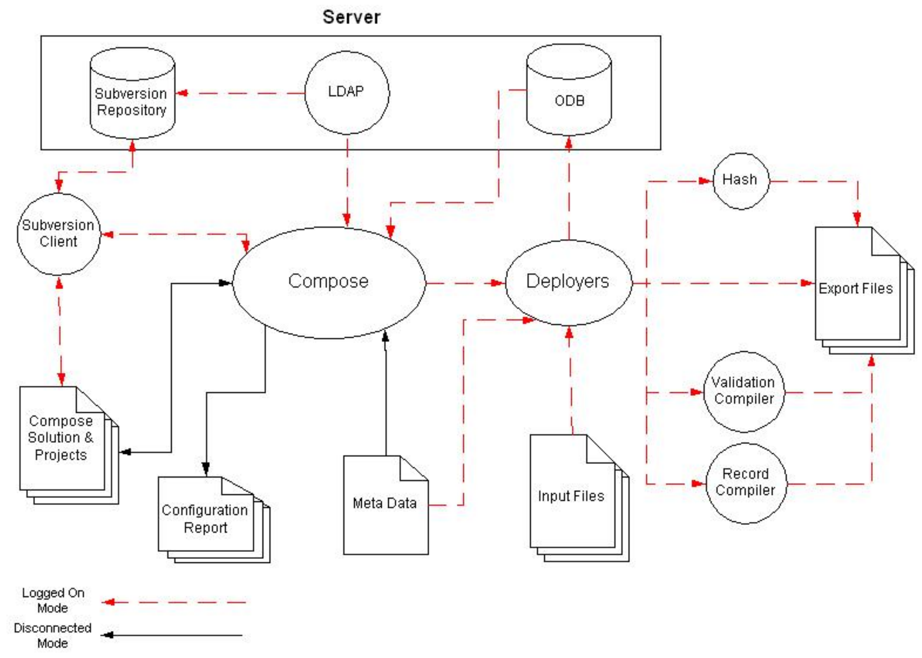
[Picture]
For more details see: PickList Editor Architecture

12.7. WA



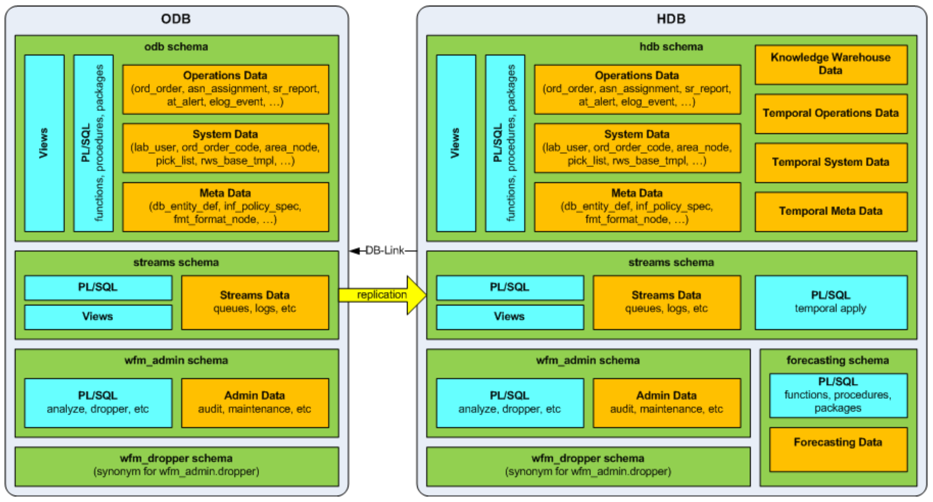
[Picture]
For more details see: WA Architecture

12.8. Compose



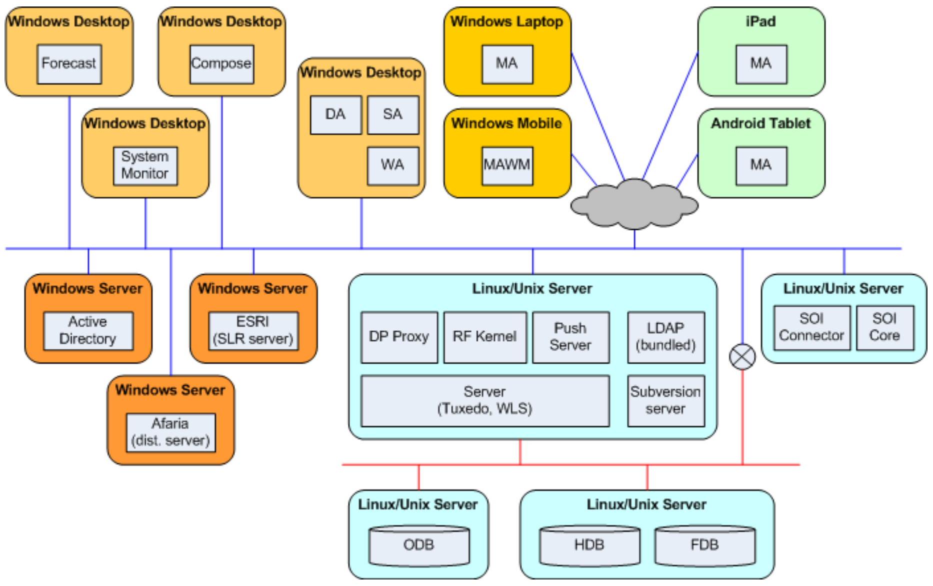
For more details see: Compose Architecture

13. Database Architecture


ODB
odb schema
Views
PL/SQL functions, procedures, packages
Operations Data (ord_order, asn_assignment, sr_report, at_alert, elog_event, ...)
System Data (lab_user, ord_order_code, area_node, pick_list, rws_base_tmpl, ...)
Meta Data (db_entity_def, inf_policy_spec, fmt_format_node, ...)
streams schema
PL/SQL
Views
Streams Data queues, logs, etc
wfm_admin schema
PL/SQL analyze, dropper, etc
Admin Data audit, maintenance, etc
wfm_dropper schema (synonym for wfm_admin.dropper)
DB-Link
replication
HDB
hdb schema
Views
PL/SQL functions, procedures, packages
Operations Data (ord_order, asn_assignment, sr_report, at_alert, elog_event, ...)
System Data (lab_user, ord_order_code, area_node, pick_list, rws_base_tmpl, ...)
Meta Data (db_entity_def, inf_policy_spec, fmt_format_node, ...)
Knowledge Warehouse Data
Temporal Operations Data
Temporal System Data
Temporal Meta Data
streams schema
PL/SQL
Views
Streams Data queues, logs, etc
PL/SQL temporal apply
wfm_admin schema
PL/SQL analyze, dropper, etc
Admin Data audit, maintenance, etc
wfm_dropper schema (synonym for wfm_admin.dropper)
forecasting schema
PL/SQL functions, procedures, packages
Forecasting Data

- **ODB**: Operational Database, contains all the data for the day to day operations. Accessed only by Service Suite components. Data no longer needed, such as completed/cancelled orders, inactive system data, etc, is retained for a brief period (to allow guaranteed replication to the HDB) and then purged.
- **HDB**: Historical Database, contains historic data. Used for reporting and analytics. Further divided into:
  - replicated data: a copy of the latest/last version of the operational
  - temporal data: a copy of each version of the operational data
  - knowledge warehouse: aggregated data specifically for reporting
  - forecasting database: forecasting data used for analytics

14. Deployment



[Picture, Source]

The above diagram shows a typical deployment.

- The Service Suite server, ODB and HDB are deployed on separate Linux/Unix servers.
- A separate LAN is used between the server and the DBs.
- External high performance storage is used for file systems with high I/O rates (eg. DB files, queues, logfiles, etc).

15. Third Party Software

The Service Suite server runs on:

- RedHat Linux (x86-64)
- Solaris (SPARC)
- AIX (POWER 7)
- VMware (Appliance - Oracle Enterprise Linux 5.8)

The Service Suite Clients run on:

- Windows 8
- Windows 7
- Windows Mobile 6.5 (MA only)
- iPad with iOS 7 or above (MA only)
- Android Tablet with Android 4.3 or above (MA only)

Service Suite server heavily relies on the following third party software:

- Oracle Database (Enterprise Edition) 11gR2
- Oracle Weblogic Server (WLS) 12c
- Oracle Tuxedo 11g

For a complete list of the third party software used see: Third Party Software (Service Suite)

16. Configuration Management

Service Suite uses ClearCase UCM for configuration management. GIT was used to develop Tablet Mobile.

17. Significant Changes from Previous Major Release

The significant changes from 9.2 are:

- Tablet Mobile for iPad and Android tablets using Sencha Touch + Cordova
- Removed the 9.2 version of MA Handheld Edition that has iPhone and Android phone support
- Service Suite Appliance and deployment in a VMware ESX 5.x environment.
- Hyperic based System Monitoring for the Service Suite Appliance.

The significant changes from 9.1 are:

- Corba removed from the server. All Tuxedo remote calls are now ATMI.
- Crew role removed. Crews are now represented by a mobile user (the crew leader).
- MAHE for iOS.
- MAHE for Android.
- MAWM (aka MA-PPC) revived.
- Scheduling Only solution. Service Suite is used for scheduling only. Mobiles are connected to an upstream system and order progression is performed through the host interface.
- Supervisors on DA: Supervisors may have limited access to the DA.
- Hierarchical area: Areas can be arranged and display in one or more groups.
- VM and Service Suite Management Console. Prototype for Service Suite appliance deployment.
- Added Forecasting feature.
- Added support for Work Orders.

The significant changes from r8 are:

- New order model. Orders now consists of one or more activities. Data is split into business rule data, extended (opaque) data and state. The extended data replaces the adv_ fields. The server handles opaque data as XML blocks (rather than individual fields).
- New implementation of the order model in Java. The bulk of Ord, Asn and Lab has been re-implemented in Java to support

the new order model. The structure of the implementation is changed to services (respresenting business functions), operations (representing actions on entities), and entities (representing data objects).

- Consolidation and standardisation of the host interfaces. Most host messages have been changed to a "verb" and "nouns" structure. The same message is used for different (but related) actions or events (indicated by the verb), and it contains the nouns (data) appropriate for the verb. Data is structured (not flat). Configuration is at the verb, noun and group level (not individual field). Compound messages are supported (eg. create and assign, modify or create, etc). Messages are further standardized to SOI canonical messages by the SOI Connector.
- Integration of WO and AB. Appointments are now scheduled by WO (as skeleton orders). OssApp is eliminated.
- Compose has been redesigned to support multiple simultaneous users and to handle the new order model and host messages. It uses a Subversion repositiory to version-control the configuration.

18. Appendix

18.1. Notes

18.2. Suggested and Pending Page Revisions

- The Diagram in section "System Context"
  - should have a dashed line from DA to ODB, to show that the DA does some 2-tier operations.
  - should have SOI added.

18.3. References

18.3.1. Attachments

Name	Size	Creator	Creation Date	Labels	Comment
network_computing.png	125 kB	William Cheung	Mar 03, 2014	None	
ServiceSuite9.3_Arch_v1.6.vsd	1.18 MB	William Cheung	Jan 03, 2014	None	Service Suite 9.3 Architecture (TOGAF)
AppComm.png	68 kB	William Cheung	Jan 03, 2014	None	
ServiceSuite9.3_TRM_v1.5.png	152 kB	William Cheung	Dec 20, 2013	None	
vss_arch_data_flow.png	23 kB	William Cheung	Dec 20, 2013	None	
vss_arch_uc_config.png	63 kB	William Cheung	Dec 20, 2013	None	
vss_arch_uc_dispatch_wo.png	77 kB	William Cheung	Dec 20, 2013	None	
vss_arch_uc_crew.png	48 kB	William Cheung	Dec 20, 2013	None	
vss_arch_uc_mobile.png	53 kB	William Cheung	Dec 20, 2013	None	

vss_arch_func_decomp.png	95 kB	William Cheung	Dec 20, 2013	None	
ServiceSuite9.3_TRM_v1.1.png	145 kB	William Cheung	Nov 14, 2013	None	
Architecture Diagrams.vsd	702 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
System Context.png	73 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
ApplianceInterface.png	58 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
Server Architecture.png	30 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
Deployment.png	39 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
Internal Interfaces.png	44 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
External Interfaces.png	37 kB	Goran Jovanovic	Oct 11, 2013	None	Whitewater changes
mahe2.vsd	70 kB	William Cheung	Oct 10, 2013	None	Service Suite Mobile Architecture source
mahe2-arch-detail.png	30 kB	William Cheung	Oct 10, 2013	None	
mahe2-arch-summary.png	34 kB	William Cheung	Oct 10, 2013	None	
Service Suite Appliance.png	25 kB	Goran Jovanovic	Feb 19, 2013	None	Whitewater changes
Database Architecture.png	50 kB	Goran Jovanovic	Feb 19, 2013	None	Whitewater changes
MA-HE-iOS-native.dot	3 kB	Goran Jovanovic	Feb 19, 2013	None	Graphviz format of the MA-HE iOS native diagram. It is partially generated by objc_dep.py. The file can be viewed by using GraphViz (freeware)..
MA-HE-iOS-native.png	1.05 MB	Goran Jovanovic	Feb 19, 2013	None	MA-HE iOS Dependency Graph
MA-HE-iOS-arch.png	46 kB	Goran Jovanovic	Feb 19, 2013	None	MA-HE (iOS) Architecture

client framework.png	60 kB	Goran Jovanovic	Feb 19, 2013	None	Client Framework
System Context Picture.png	585 kB	Goran Jovanovic	Feb 19, 2013	None	Obsolete: From the Functional Spec; source available from ProdMgmt
ServiceSuiteAppliance.png	25 kB	Goran Jovanovic	Feb 19, 2013	None	Obsolete

18.3.2. Descendants

- Server Architecture (Service Suite)
- Service Suite Appliance Sizing

18.3.3. Incoming Links

The license could not be verified: License Certificate has expired!

18.3.4. Pages with Architecture and Service Suite Labels

External Interfaces (Service Suite)

Host Interface (Service Suite)

Internal Interfaces (Service Suite)

Server Architecture (Service Suite)

Server Architecture (Service Suite) - 9.2

Service Suite Architecture 9.2

Service Suite Architecture 9.3

Tuxedo Server Components (Service Suite)

Tuxedo Server Processes (Service Suite)

Category Architecture, Category Service Suite

architecture service_suite favourite menu_pd