

StoneOne - The Web Service Factory

Technical Whitepaper



Consider Platform as a Service (PaaS) in your Cloud Strategy

"Gartner predicts that by 2015, most enterprises will have part of their run-the-business software functionally executing in the cloud, using PaaS services or technologies directly or indirectly. Most such enterprises, will have a hybrid environment in which internal and external services are combined."

Gartner, Inc., 2011



© Copyright 2008-2011 StoneOne AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of StoneOne AG. The information contained herein may be changed without prior notice.

Some software products marketed by StoneOne AG and its distributors contain proprietary software components of other software vendors.

Apache Tomcat and Tomcat are registered trademarks of the Apache Software Foundation

Hibernate is a registered trademark of Red Hat, Inc.

IBM Lotus Notes is a registered trademark of IBM Corporation in the United States and/or other countries.

Java is a registered trademark of Sun Microsystems, Inc.

Microsoft, Windows and Exchange are registered trademarks of Microsoft Corporation.

OSGi is a registered trademark of the OSGi Alliance

SAP is a registered trademark of SAP AG in Germany and/or other countries.

Spring Framework is a registered trademark of SpringSource, Inc.

XML is a registered trademark of W3C, World Wide Web Consortium, Massachusetts Institute of Technology.

ZK is a registered trademark of Potix Corporation

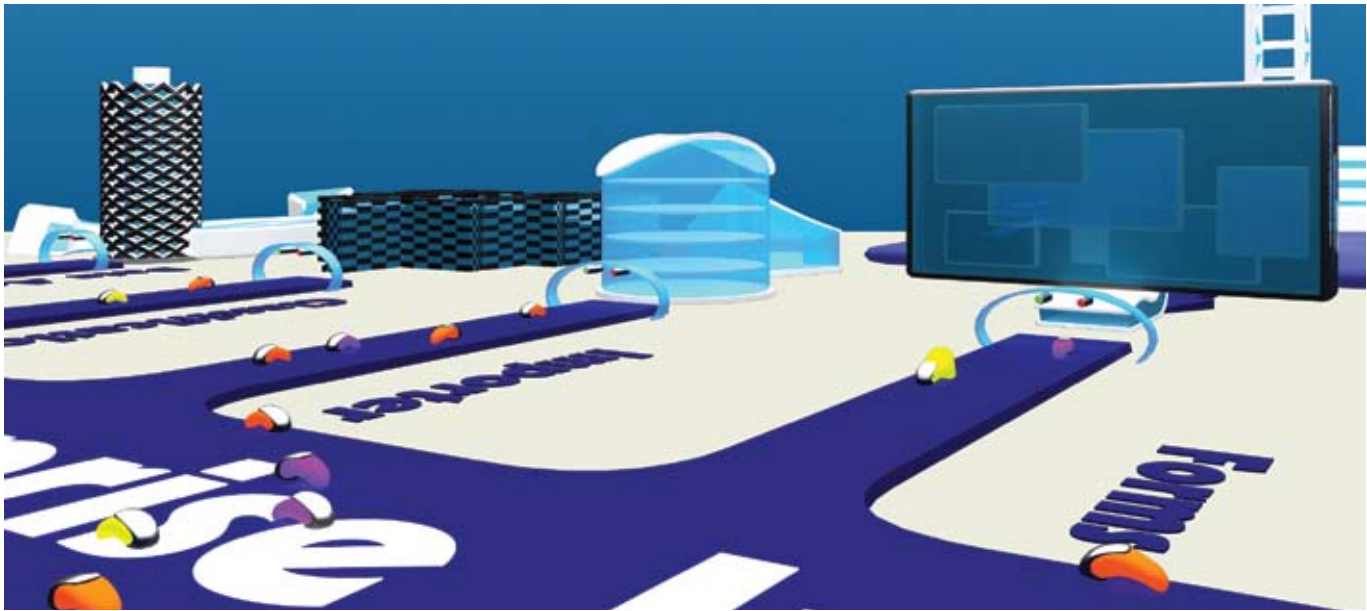
All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by StoneOne AG and its affiliated companies for informational purposes only, without representation or warranty of any kind, and StoneOne shall not be liable for errors or omissions with respect to the materials. The only warranties for StoneOne products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.



Contents

Introducing StoneOne Web Service Factory.....	1
Business and IT Advantages.....	1
Platform as a Service: Integration, Processes, Documents.....	2
Web Service Capabilities and Use Cases.....	4
Typical Use Cases.....	5
Strategic Advantages.....	5
StoneOne Web Service Factory Product Overview.....	6
Design and Development.....	7
Enterprise Bus Architecture.....	7
Components.....	8
Application Connector.....	9
Application Designer.....	9
Audit/Logging.....	9
Billing&Tracking.....	10
Business.....	11
Classification.....	11
Cloud Extender.....	12
Communication.....	12
Data Acquisition.....	12
Data Pattern (Data Model).....	13
Exporter.....	13
Extraction.....	13
Importer.....	14
Monitoring/Maintenance.....	14
Reporting.....	15
Rules/Logic.....	15
Storage.....	15
User-Administration.....	16
Runtime Environment.....	17



Introducing StoneOne Web Service Factory

It is our aim to enable our customers and partners to gain access to the world of web-based business services as easily and cost effective as possible. StoneOne created a completely new software generation featuring standardized components for individual business services – in the web and also in the cloud. State-of-the-art web solutions perfectly answer the customers' need for modern, interactive support of their business processes. Web-based business solutions can map entire processes as a service in the web browser, while accruing no costs for implementation or administration and offering greatly reduced operating costs. Modern web solutions utilizing current bus architectures can also easily access additional resources in various clouds depending on demand. Users can access the most up-to-date information, whenever and wherever they are, and the only maintenance work required is performed at the central server of the service provider.

However, the currently used methods of creating web business solutions are quite complicated and require a lot of maintenance because they are based on individual design and programming and the application of extensive tool sets.

StoneOne addresses these challenges through an extensive bus architecture based on an Enterprise Service Bus (ESB) in combination with huge set of components for multiple purposes and a configurable data model. Since the StoneOne architecture goes beyond ESB and contains several components focused on documents and information handling we call our solution Enterprise Information Bus (EIB). The EIB is mostly focused on Software as a Service (SaaS) and Cloud enablement – the main features are platform as a service (PaaS) enhanced by some Infrastructure as a Service (IaaS) logic together with a set of SaaS standard features.

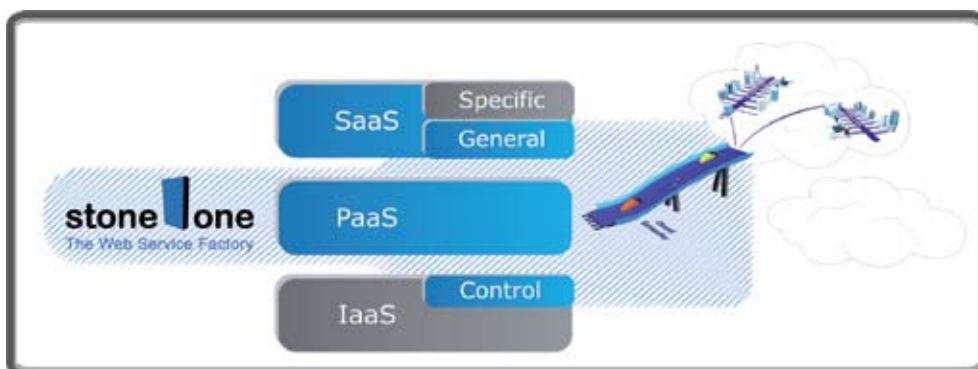


Figure 1: Positioning of StoneOne's Web Service Factory



The StoneOne Web Service Factory combines a unique technological platform (software factory) and methodology to reduce the cost and risk involved in developing individual business solutions. The factory can be used as a whole but also only in the required parts – the components. Also, thanks to using standardized components, the services are easily customized to meet the spe-

cific needs of each customer. Applications already existing in the customer's IT landscape can be integrated using the service components (application connectors) provided by StoneOne. StoneOne offers its customers a tailor-made solution that minimizes project risk and can be adapted to the needs of many industries.

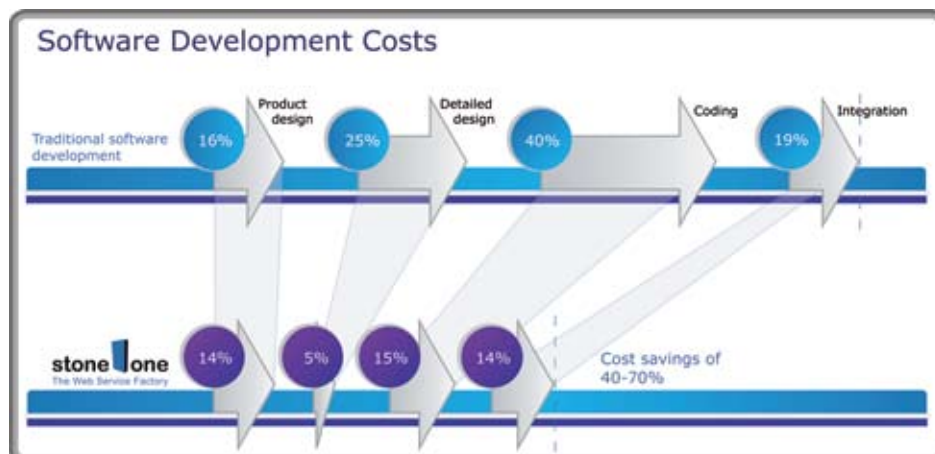


Figure 2: Time and cost reduction in development of business portals and services

Business and IT Advantages

The Web Service Factory is designed to address a range of high priority software requirements of most enterprises and many software manufacturers. Specifically these requirements involve the following:

- Integrating the various software applications needed for the core business
- Making core business processes more efficient
- Migrating legacy software to the web
- Rapid prototyping: Faster and risk-reduced implementation based on SOA

Platform as a Service: Integration, Processes, Documents

Platform as a Service (PaaS) combines older technologies (legacy software) into a single platform to achieve a robust and reliable application building process. It combines central elements out of EAI, BPM, DMS, ECM, ASP and Software On-demand into a scalable platform.



Typical Business Environment

Today's enterprises utilize diverse set of multiple applications for doing their business. When implementing efficient processes integration becomes a critical aspect because of the need of using necessary functions of the current software infrastructure. The highly flexible architecture and data-models provide strong capabilities for integration of external functions and accessing external data models.

Processes

Many processes in today's enterprises are still only partially automated, especially when working with distributed clients and unstructured documents, i.e. a highly distributed organization of small units, many single clients or resulting out of M&A activities. By implementing efficient central services the process cost per transaction can be reduced dramatically.

Documents as Part of Most Processes

Especially distributed processes often require supporting documents and/or images as part of processes. When these documents are part of the electronic process from the very beginning strong automation gains can be achieved.

Migration to Web x.x

The Web Service Factory allows for a flexible approach to migrate existing business applications to a modern design and architecture. New web solutions can easily designed by StoneOne's EIB and it's components utilizing legacy applications in the background. This way an existing software base can be migrated to a SaaS/PaaS solution step-by-step.



Web Service Capabilities and Use Cases

StoneOne Web Service Factory is a product suite that can be applied to various business processes. The essential principles behind Enterprise Bus scalability – illustrated in the diagram as below – are:

Standardized components for individual business processes save time and money. Ready-made building blocks (components) from the software factory significantly streamline the process of developing business portals and can be combined to taste.

The standard components from StoneOne as well as specific components from partners can be used separately or in combination with existing software. They can be combined by using an Enterprise Bus architecture like ESB (Enterprise Service Bus) or StoneOne’s EIB (Enterprise Information Bus) can be used to **centrally control all components** of the software factory.



Figure 3: StoneOne’s Key Capabilities



Typical Use Cases

Examples of typical business challenges and representative solutions are

- Front-office/back-office integration
- Processing multi-step transactions across multiple systems
- Self-service applications, i.e. customer or employee self services
- Human-centric workflow processes
- Integrated view across applications (improved management information/overview) leads to efficiency improvements,
- Enrich existing software with crucial components like Billing&Tracking or Rules/Logic

Strategic Advantages

Examples are

- Support for IT integration in distributed environment, i.e. caused by M&A
- Real-time overview across applications
- Much better TCO
- Faster deployment, rapid prototyping
- Easier roll-out, update, maintenance



StoneOne Web Service Factory - Product Overview

The StoneOne Web Service Factory includes tools and a set of components for designing, creating and running portal-based business solutions. The

key components are illustrated in the diagram as below:

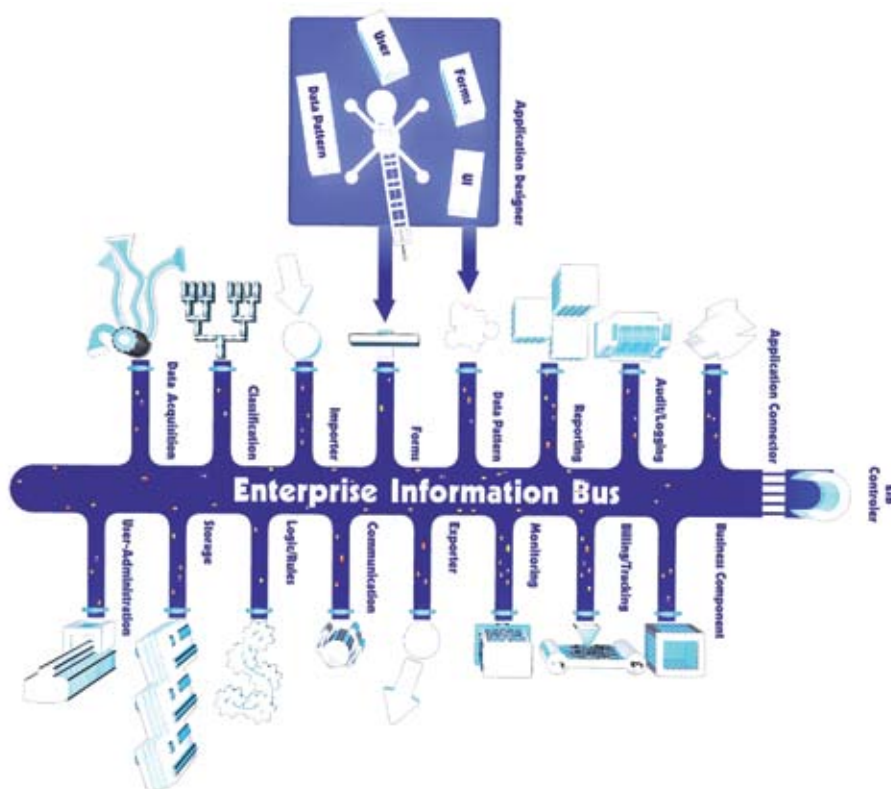


Figure 4: Enterprise Bus Architecture (ESB/EIB) with components



Design and Development

Business applications are designed by using the “application designer”, which allows for client-side definition of user interface, data models and configuration or scripting. This way a business application is defined which is later on generated and then deployed to the Bus Server which will unpack and execute the new application. The defined business application utilizes the various standard components together with integrated legacy systems and newly developed business components to form a new application.

StoneOne’s EIB and standard components use established de facto standards like Hibernate, JMS, OSGI or Spring to provide very strong functionalities for the generated applications.

Application Design/Rapid Prototyping

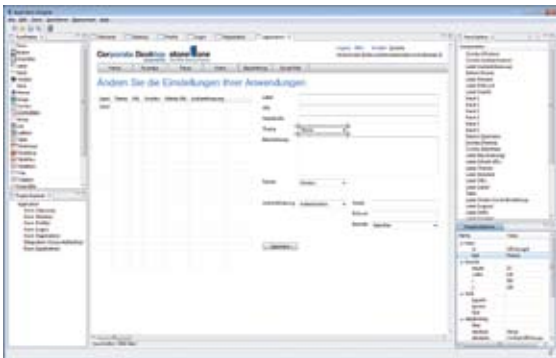


Figure 5: Application Designer/Form Pattern

The core elements of a business application are defined by a simple definition of all tables which are part of the data model combined with a set of form patterns, which describe all aspects of the user interface together with role-based access rights.

Enterprise Bus Architecture

If an “all-in-one” solution has to be build, the optional “Enterprise Information Bus” (EIB) from StoneOne is used to centrally control all components of the software factory. The EIB is a logical extension of an Enterprise Service Bus (ESB) adding new features and capabilities.

The StoneOne technology was designed for ease of administration and solid performance in heterogeneous system landscapes with high data traffic. All components relevant for electronically mapping and optimizing business processes – including existing applications – can be managed and controlled using the bus architecture.

The Enterprise Bus represents the central and comprehensive controlling system of StoneOne’s software factory. The Bus is important when developing complete software solutions, it is one efficient way to connect all components and manage scalability. The accumulated experience of hundreds of projects, many of them very complex ones, was shaped into an architecture that combines the virtues of simplicity, easy handling and versatility, while at the same time performing even under extreme load.

Based on modern, service-oriented architecture, StoneOne offers a powerful enabling technology for the fast development and implementation of B2B and B2C business services.

The key strengths of central control via EIB are:

- Consistent and comprehensive data structure
- Universal “backbone” for all components and the derived services
-



- Consistent policy for the cross-system description of data models of all clients and their applications
- One consistent design
- Common interface and communication behavior of all components
- Uniform maintenance interface
- Central controlling of scaling and load behavior of the different components

Components

Standardized components combined with a standard procedure/methodology for building individual business processes save time and money. Ready-made building blocks from the software factory significantly streamline the development process and can be combined to taste.

One of StoneOne's key advantages are the ready-made standard components already existing for many processes, such as for data acquisitions (uploading documents, images), designing forms, modeling databases/tables, billing/tracking, rules/logic, logging or user administration. This allows StoneOne to offer its customers a very cost-effective solution as the development efforts are much lower compared to an individual project solution.

The standardized business components implement frequently recurring business logic or functionality and can be employed in a broad range of vertical or horizontal web solutions based on the StoneOne technology.

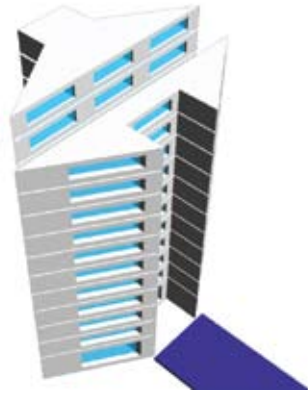
Together with the well-defined Enterprise Information Bus (EIB) system interface, the components draw on fixed and standardized data, metadata and interfaces and are connected by a rule engine

to form a rule-based technical workflow. Each StoneOne component also has a status/maintenance interface, which greatly simplifies administration in web-based software solutions.

StoneOne provides a generous set of standard components which simplify the building of individual business applications by reducing the amount of individual implementation work and also by providing a standard concept of how to implement these business solutions.

These components are:

- Application Connector
- Application Designer
- Audit/Logging
- Billing&Tracking
- Business
- Classification
- Cloud Extender
- Communication
- Data Pattern (Data Model)
- Exporter
- Extraction
- Importer
- Monitoring/Maintenance
- Reporting
- Rules/Logic
- Storage
- User Administration

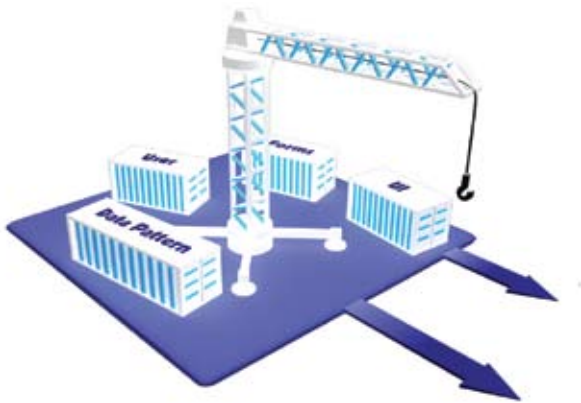


Application Connector

The StoneOne business services very frequently interact with a wide variety of standard applications. For this reason, specialized connectors establish a connection to selected functions of standard software packages, such as SAP, Microsoft Exchange, Lotus Notes or even connect with specific applications of partners or customers.

The “application connector” is a kind of wrapper for an external application which typically implements only the required interface parts for EIB usage.

As a result, these “wrapped” functions can be used without additional development in any kind of configuration, i.e. as part of a form-pattern checking formula or to implement external credit-limit-checks.



Application Designer

The “application designer” contains most of the elements necessary to design an individual business portal or web application. It allows for the design of complex individual data models,, design of user admin/multi tenancy and design of forms.

Many applications require thousands of individual data elements to allow for mapping all possible types of forms.

Main elements are the data patterns, which describe a data model per application which can consist

of multiple tables with complex relationships. These data models can be used transparently as part of any configuration and allow for easy access.

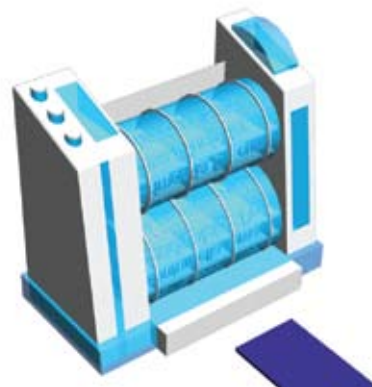
Very important is also the form pattern, which describe very flexible user interfaces in a general way which can later-on be translated (rendered) into the specific environment (supported user interface frameworks like ZK, JSF/RF, etc). These form patterns support the design of nearly any kind of business user interface in combination with business logic like command buttons, search, result lists, menu’s, tab menus, images, etc. to create a well designed user interface ranging from simple to highly complex applications. Style sheets for simplifying the design are also supported.

This facilitates the fast and efficient creation of entry screens for business cases. Prototypes and customer-specific scenarios are designed with ease.

Deployment/Update (of Applications)

Any application developed needs to be rendered in an environment. Deployment is currently prepared by the application designer and packages the application as required by the specified environment. The upload of this package to the EIB-server is done through the selected application server, it extracts and activates the package – finally making it accessible to the user.

The StoneOne platform offers convenient provisioning, help and support functions for the initial implementation of a designed application and all its parts and for introducing updates.



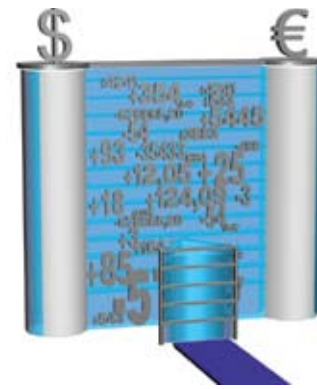
Audit/Logging

The StoneOne platform ensures that all system events are stored at the file and database level at a configurable access level. Audit tables recording application-specific actions or processes required by the customers can also be configured.

An audit log is a simple but very effective form of

tracking temporal information. The idea is that any time something significant happens; some record is written indicating what happened and when it happened.

“Audit/logging” is the process of managing the audit logs on a particular event and entity of the system. As an example, in an application, every attempt to use the “storage” component can be logged in the audit log. This way, the application is saved from writing auditing code internally – the internal details of how auditing is achieved are hidden from other parts of an application.



Billing/Tracking

One of the StoneOne platform’s key features is a multi-level billing system with integrated logging and communication. The “billing” and “tracking” component is a component that helps manage and automate billing and tracking process in the system. Billing and tracking is the money-making component.

Billing strategy is the mechanism to define the payment plan and invoicing scheme for each customer as per his requirement. The strategy can be of multiple layers. The payment scheme can be varied on different layers of the strategy.

Billing is the process of processing and rendering a bill/ invoice.

Tracking is the process of collecting data to enable monitoring of resource usage.

The component tracks the usage of specific features of other components and generates invoices based upon billing strategies.

There are configurable counters for all relevant system results which allow for accurately tracking and billing the use of specific functions. For example, you could attach a fee to storing an object, which is then automatically billed either on a monthly basis or after reaching a predefined number of uses. Specifically, a report is generated which can be printed and sent by mail or immediately delivered by e-mail. The corresponding invoice can be directly based on this report.



Figure 6: Audit Report

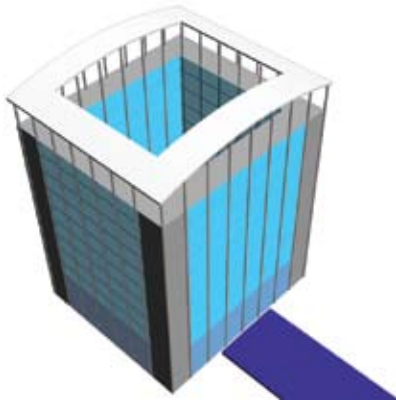
The design is intended to address the following goals:

- Generalized way to implement audit logging that can be used to provide audit logging needed by all parts of an application.
- Achieving the consistency to log every occurrence of an event that needs auditing.
- Allows handling of customized EIB audit events.



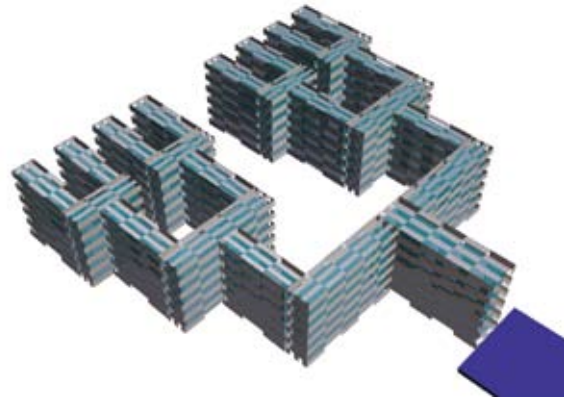
Figure 7: Billing Policy

For example: Organization A may charge an amount of 1 € per component usage to organization B and B can charge an amount of 1.5 € per component usage to organization C and 2.0 € to organization D. This is called a multi layered strategy and the invoicing needs to be generated separately as per the strategy payment plan to each customer.



Business

The "business" component is normally developed by specialized partner companies or specialist departments of the customer. In either case it is based on the respective expert knowledge – StoneOne delivers this component as an "empty vessel." It delivers a specific, complex functionality that cannot be implemented as part of a configuration process. The corresponding functions are mapped by the StoneOne technology as standard functions to make them available for use with, e.g., the "workflow" or "forms" component. A simple example could be implementing comprehensive data checks that cannot be mapped on a field level.



Classification

Incoming data can be analyzed and allocated to a specific topic (e.g. invoice, delivery note, advertisement etc.) with the "classification" component in combination with the rule engine. Specialized software programs for intelligent text analysis and extraction are also integrated.

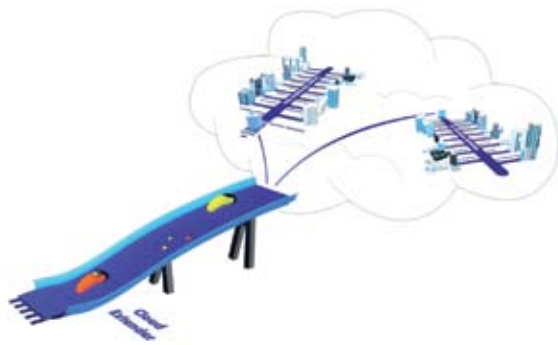
EIB classification component is one answer to the problem of classifying data ranging from e-mails to invoices to excel sheets and so on. It can classify any file or object to its respective category without human intervention. It effectively manages large volume of data in fraction of seconds.

It provides the capability to define standard classification schemes that classifies any object or files. Classification schemes refer to the set of rules using which classification is performed.

The classification component will integrate with any of the available classification systems.

The design is intended to address the following goals:

- It uses an administration interface to add classification schemes/rules to the training engine. These rules classify the data into their respective category.
- This interface can also integrate with any of the classification systems.
- The incoming data can be XML with meta-data. Meta-data will be used to classify the data object.
- It generates the EIB data object as an output.
- It will be capable of performing simple pattern recognition without using OCR by identifying simple rules.
- If classification component cannot classify then administrator interface will provide the queue of unclassified data.



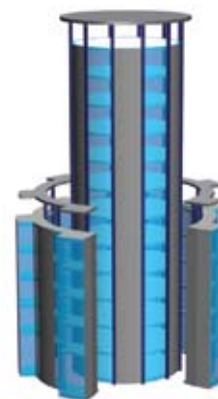
Cloud Extender

This component is a functional extension of the Web Service Factory for expansion into a cloud environment, it connects to different cloud providers like Amazon or IBM depending on configuration. Each server runs one or several EIB instances which are administered by their own EIB controller. Several instances are connected through their respective controllers – each instance may contain different components for specific use-cases. One of the instances hosts the master controller, which will upon request book resources in a selected cloud and connect to the new instance. The cloud extender uses pre-deployed virtual systems, to reserve resources, connect the resources with the required virtual system and establishes a connection. Afterwards the bus controller will measure performance in each distributed system and will delegate tasks accordingly.

An EIB Controller administers the queues and transaction times of its EIB instance, if pre-configured thresholds are met, it will send an alert to the master controller. The master controller is aware of all active instances and their capabilities and can redistribute the work to other instances. The Cloud Extender is the specific interface for the concrete cloud provider, depending on policies including budgets and thresholds the master controller may book additional resources, activate instances and distribute the workload.

Administration and control is done by policies, depending on parameters like “very good response time”, “guaranteed response time” and “budget limits”. In case of shrinking demand the master controller can also close and deactivate resources in the cloud.

Cloud in this context can be several different technical implementations – it ranges from public clouds like the Amazon cloud over hybrid cloud to private clouds based on systems like VMware or Virtual Box.



Communication

This component provides standard routines for any kind of usage in designed applications like

- sending e-mail,
- printing text, documents, reports,
- SMS for short but fast notifications etc.

to allow i.e. the automatic sending of a report or the notification of an administrator in case of emergency.

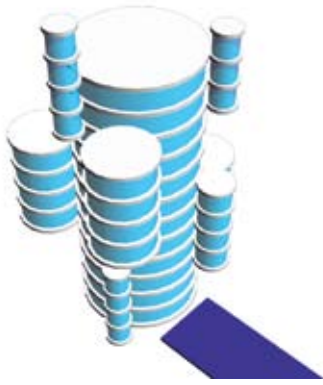


Data Acquisition

The useful data, especially electronic or scanned documents (invoices, orders, travel expenses), are transferred from the end user to a data queue on the EIB server via data transfer. Multiple queues can be created for each client. A rule-based analysis, which draws on the classification component, turns this into a data object which is connected to a data pattern as well as a form pattern for each client. All data is transferred in a transaction-safe manner, when possible broken connections will restart near the breakup-point.

“Data acquisition” component also generates metadata defining certain details or metadata of data/file uploaded.

During upload, data acquisition shows a progress bar to inform user how much volume is left before the upload is complete.



Data Pattern (Data Model)

Multiple data patterns can be stored for each tenant. They support the representation of extensive relational structures and can also be linked to audit functions. A state-of-the-art administration interface (see "application designer") is provided for this purpose.

The resulting "database" models and functions are provided transparently for all other components and are transaction safe for all common databases.

Database designing is the process of designing the complete structure of the database that includes

- Selection of database technology (MySQL, Sql Server, Oracle, Cache, Lotus Notes etc).
- Database properties (like name)
- Database tables
- Fields of the table
- Constraints – like primary key, foreign key etc
- Relationships between database tables.

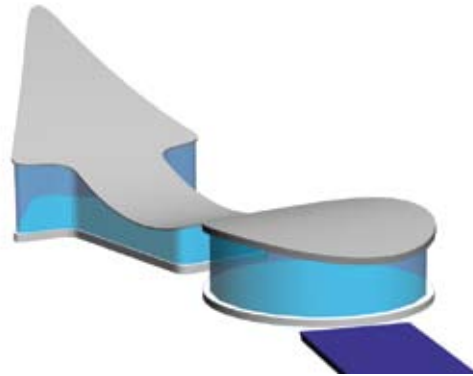
The database designer is a graphical user interface to design the structure of the database. The output of the database designer will be an XML (data pattern) that will be used by database component to create the actual database.



Figure 8: Database Designer

Data Pattern component will be used to create the database with all its objects like tables, data fields, and the relationships between database tables based on the data pattern generated by the database designer.

Database component will be used for CRUD (create, read, update, delete), import, export and other advance operations (like transaction management, batch updates etc).



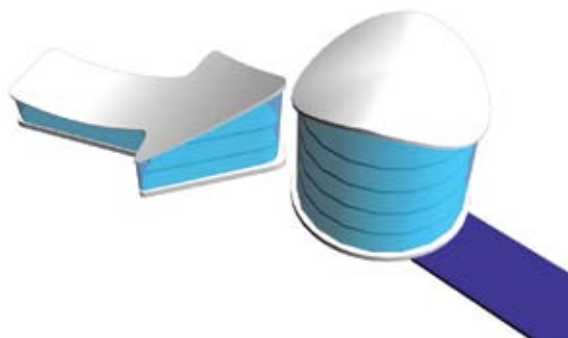
Exporter

The "exporter" is used to transfer tables in CSF format to local systems for further processing (e.g. to perform special evaluations).

The export component can export all table, queries and views (from EIB system) in the requested format to the authorized user.

Extraction

This component is part of the data acquisition technical workflow. An incoming file object is put into a tenant specific input queue, depending on queue parameters extraction is part of the analysis process. Incoming file objects are analyzed, if the file type respectively content is known, the component will extract available metadata like file properties and also the content of the file. Optionally it is also possible, to attach an OCR systems for text extraction out of images and also, to extract positional information for each word if it is a PDF/A document.



Importer

The “importer” is used for the transaction-safe import of bulk data (e.g. address data of a client) into tables of a usage scenario. The data is imported by block, and in the event of an interruption, the process is resumed intelligently.

The import component will provide an extremely fast and stable set of classes for parsing virtually any commonly found delimited data format and import the data directly into the selected data pattern.

During import, it shows a progress bar to inform the user of how much of a total task (in terms of size) has completed.



Monitoring/Maintenance

“Monitoring” represents the graphical interface to the maintenance interface of the components, it helps to monitor and maintain all the components in the system. The status of each component is indicated as “red”, “yellow” or “green”. Detailed information is available on the maintenance history, including the results, an administrator comment and the status of the final result. The component may also indicate the maintenance tasks to be carried out in the very near future.

This procedure ensures that the operation history is fully documented.

“Maintenance” is the process of enhancing and optimizing the components in the system.

It is the care and servicing of the system to be in operating condition by providing systematic inspection (health check), detection, and correction of failures either before they occur or before they develop into major defects.



Figure 9: Monitoring Overview

Maintenance is needed to:

- Correct errors
- Correct requirements and design flaws
- Make enhancements
- Inform administrator proactive

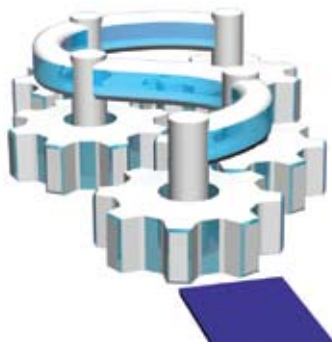
Monitoring means to be aware of the state of a system. The use of conditional monitoring (process of monitoring a parameter of condition) allows maintenance to be scheduled, or other actions to be taken to avoid the consequences of failure, before the failure occurs. Action is taken to find and eliminate the root cause of the problem and, if possible, bring the required data from EIB and bring back the component to an acceptable operating condition.



Reporting

The component uses defined data-models to create reports in different formats. Each report contains a set of records based on selection criteria and a template for the desired report.

The report templates can be configured and can also contain graphic elements which could be used



Rules/Logic Engine

The “rules/logic engine” component uses its own rule engine to implement rule-based workflows. A rule component is a software system that helps to manage and automate business rules. It applies predefined rules on data and takes the appropriate action.

The rules may come from legal regulation („an employee can be fired for any reason or no reason but not for an illegal reason”), company policy („all customers that buy more than \$100 at one time will receive a 10% discount”) or other sources.

The rule component software, among other functions, may help to register, classify and manage all these rules, verify consistency of formal rules, infer some rules based on other rules, and relate some of these rules to enforce one or more of the rules.

Rules can also be used to detect interesting business situations automatically. For example, „notify sales when inventory is lower than 10 and we have

more than 5 pending orders on a monday.” This component will provide all rule specific workflow and action logic.

Rules-/forms-based workflow which is supported as part of EIB is targeted for simple document or information based processes like signature or their way of through organizations. More complex and elaborate workflow processes will be implemented



Storage

Almost every large application needs storage management solutions for storing and persisting data. These solutions have become more important as the amount of information that companies need to store continues to grow.

The “storage” component makes sure that data is archived in auditable form, be it on RAID, SAN, EMC Centerra, NetApps or using tape or jukebox systems.

Also, any commercially available archive system can be connected using standards (e.g. CMIS, JSR-170, JSR-283) or, if necessary, based on a customized solution.

EIB storage component will allow the storing of EIB data objects in any kind of storage media. Any significant change in an object will lead to updating the current object in storage, where the old object is still available through a back-link.

Storage component will also have a mapping capability which will map EIB data object ID with storage ID.

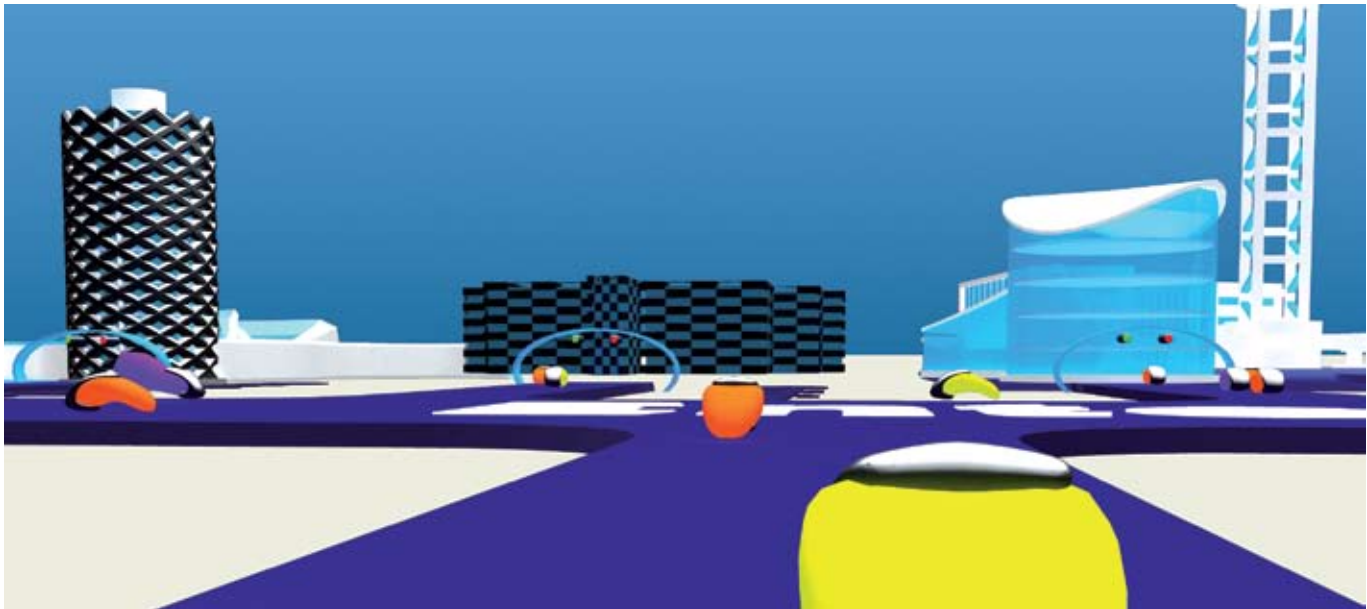


User-Administration

“User-administration” is based on a so-called proxy that centrally manages all available user information across all tenants and can also store additional information (rights, profiles, etc.) as schema extensions. The proxy further supports automatic synchronization with client systems.

This component also supports role-based rights/permissions for users, mapping for roles and single sign-on.

Profiles/start forms can be assigned to groups and also to users depending on the level of portal integration.



Runtime Environment

The primary runtime environment of StoneOne's Web Service Factory is a JEE compliant server, that provides the basis for EIB and which hosts the various components.

Enterprise Bus Architecture (ESB/EIB)

The central bus capabilities are:

- Messaging, messaging queues based on JSR-9141
- Component handling, loading will be based on OSGi
- Service management

The design is intended to address the following goals:

- 1. Instancing Strategy:** Instancing strategy will allow EIB, in building multiple instances of the same component. Every time an instance of the component is requested a new instance is created and returned by EIB.
- 2. Persistence Strategy:** Persistence strategy will allow EIB, in persisting instances of each component which has the necessary capabilities.
- 3. Lazy Loading:** EIB will support lazy loading of the components, this allows instances of components to be only created when they are needed. Some system components will be needed by EIB during startup and they will not be lazy loaded. But business components which are needed only at specific phase of workflow will be loaded only when, application needs to load them and till then memory will not be used.

4. Parallel Processing: EIB will allow the flexibility to spawn multiple component specific tasks which will be performed in parallel.

5. Waiting Time: The component can configure the waiting time, EIB will wait before killing the component in case of no response. In certain situation it will make sense for EIB to wait for slightly higher time for the component (batch operations) and this configuration parameter will allow components to inform EIB in case of long expected delays.

Administration Services

Presentation platform for several purposes:

- Administration
- Monitoring
- Configuration

Integration Services

Integration into existing legacy systems is provided by different application connectors. Typical connectors implement a partial or complete interface to an external application which afterwards can be used as part of the EIB standard configuration.

Rules Services

The EIB rules component can be used for implementation of any kind of rules-based process. It can evaluate multiple rules very fast, several rules can be "true" at the same time resulting in a set of actions which can be either executed directly or depending on requirements as part of a client-side logic.

¹ JSR: Java Specification Request



Monitoring and Management

A major aspect of StoneOne's EIB and components is the easy management of the system. For this reason, the monitoring component gives a comprehensive overview of the status of all components. In case of developing problems it can inform the admin actively about necessary activities. To support the daily work of administrators it also stores a maintenance history for each component which gives an overview of past problems and what was done to solve them. It can also remind the administrator of future maintenance work still to be done. Monitoring and maintenance component gives overview across all available and/or active components with current status and their respective maintenance history.



stone one
The Web Service Factory

StoneOne AG
Keithstrasse 6
10787 Berlin
Fon: +49 (0)30 469 99 07 18
Fax: +49 (0)30 469 99 07 19
info@stoneone.de
www.stone-one.com

