White Paper on

w2bill™ INVOICE

CONTEXT
ARCHITECTURE
CAPABILITIES
ROADMAP

INDEX

INTRODUCTION

If we search the dictionary for a definition of invoice we'll find something like this: an itemized bill for goods sold or services provided, containing individual prices, the total charge, and the terms.

Or like this: a document issued by a seller to a buyer listing the goods or services supplied and stating the sum of money due. However, nowadays an invoice is much more than that. In order to satisfy the growing needs and wishes of customers, fuelled by technological accomplishments, companies have to be able to adapt and innovate, all for the sake of the customer experience win-win place at the end of track.

As in various others market aspects, in billing and invoicing, the regulator struggles to keep up with the innovator, and the market player battles to comply with the regulations in place, forcing companies to incorporate innovation quickly, whilst staying compliant with the ever-changing local legal frameworks.

To survive the current global market, it is mandatory to have a robust yet flexible set of capabilities that enables companies to perform efficient and reliable changes with a quick Time-to-Market, whilst maintaining the paramount concern of charging customers properly.

CONTEXT

In almost every market, we've been witnessing consumption-based models disrupting the traditional business models. On top of that, regulatory realities are continuously changing, ever more influenced by the global economy and ways of life. Seldom do technologies or businesses remain stagnant, reliant on their modus operandi and existing capabilities. Customers, be they large-scale enterprises or the small business, are used to competitiveness and improvement and are one of the top drivers of change. In an ever more present subscription economy, with the development and wide-spread of numerous approaches such as pay-per-use, metered / provisioning / subscription billing, and other real-time, immediate billing methods, the types of services companies are able to offer to the public are becoming a critical success factor in their capture and retention of market share and quota.

All this flexibility, adaptability and agility pose an enormous challenge for companies, especially the ones with large and robust legacy systems. The key is the ability to accommodate for either (or both) batch and on-demand billing processes, in the most timely and cost-effective manner. Using solutions that are able to fit the companies' 'flavours' of billing, and in some cases couple with the existing legacy systems without disrupting their robustness or way of working, it is ensured the ability to account for a future product/service vision, future market and legal change needs in order to better compete, comply and win.

Technology constantly outpaces itself year after year, with new advances being presented incessantly, and changing the way people and enterprises do things. It continuously grows in its presence everywhere, effectively breeding a dependency like never before. People don't go anywhere without their smartphones, they operate simultaneously on their tablets and laptops, while at the same time enterprises rely on complex infrastructures to operate and effectively manage their core business. Little can be achieved without technology, and any company without a strong digital presence is doomed to oblivion.

With so much technology in place, so many different users and such diversity in services and offers, the amount of data and complexity of its management,

from a provider perspective, is ever more obvious. The evolution has been marked by moving from monolithic applications, into modular ones; from there, onto discrete domain-specific systems interconnected by complex integration architectures.

Businesses now support their flexible, adaptive, innovative and modern business models, especially on the pressure they put into operating costs. From the startup company to the large enterprise, the key to prosperity seems to be able to accommodate the markets 'mood-swings', endure the constant shifts and to compete being as light-weighted as possible.

In order to accomplish every tangible market expectation, companies must be able to change, transform and scale their supporting activities and infrastructures, all for the sake of the customer experience win-win place at the end of track.

Every aspect or dimension supporting the business-model, especially the IT infrastructure, needs to be flexible, scalable, agile, efficient and have as little implementation time as possible.

Companies seek adaptability, modularity and the ability to account for future changes. This proven fact has been creating the need for integration solutions, which can automate and scale performance in their core business support activities, diminishing the huge and lengthily headache of tying all the loose ends together with every innovation and shifting from the core legacy systems to automation and scalability with very little time to do it.

Although commonplace to everyone, most people have a basic grasp of the complexity inherent to the billing and invoicing domains, and where one ends and the other begins. People and enterprises are accustomed to leading their financial day-to-day, dealing with the fact of constantly being charged, taxed, discounted and shown a document where all is explained – to some extent. From simple "here's the total" restaurant or small-business 'invoicing', to enterprise-grade, complex, finely detailed reports and invoices, the difference between these two worlds seems non-existent.

It is important to understand that they are distinct: each one takes care of a specific set of responsibilities, but to provide a truly enriching customer experience, they must be connected. There's little gain in having a highly powerful billing engine when the information shown is outdated, unclear and unattractive; on the other hand, having an enticing document, filled with capabilities is useless when the information shown is invalid, incorrect or incomplete.

w2bill™ Invoice understands the challenges in both realities and aims to provide a consolidated response to them by:

- ensuring hierarchical account organizations, suited for even complex enterprises
- providing product catalogue management
- enabling different tax calculation
- allowing discounting per item (from the product catalogue) or the whole invoice, in either percentage or amount
- enabling real-time reporting of financial information
- providing various document types, such as invoices, credit and debit notes
- allowing document sequencing and numbering, per type
- supporting state-of-the-art document formatting and templating, from printer-specific formats to interactive web documents

CAPABILITIES

To provide a quick overview of what w2bill™ provides, a summarized list of capabilities and concepts is presented.

- **Hierarchical organization of accounts**, where the top parent account represents the actual customer, and the children are accounts that issue invoices. This approach enables a unified view per customer, whilst at the same time allowing the customer to be segmented as it desires (different departments, different geographies, different employees)
- **Chargeable items catalogue**, storing the products and/or services made available for sale
- **Tax configuration**, letting different taxation options and percentages to be defined and assigned to the chargeable items
- **Discounts**, **either of percentage or amount**, and specific to each item or to the whole invoice can be assigned per document
- **Document Types**, allowing different financial actions being carried out through specific documents
- **Realtime Reporting**, enabling visualization of organizational-relevant KPIs during the billing period
- **Document sequencing and numbering**
- **Rich document formatting options**, enabling a mix of traditional and state-of-the-art communication possibilities with the end client
- **Multi-system integration**, enabling the solution to be placed in existing enterprise infrastructures with minimal disruption

ARCHITECTURE

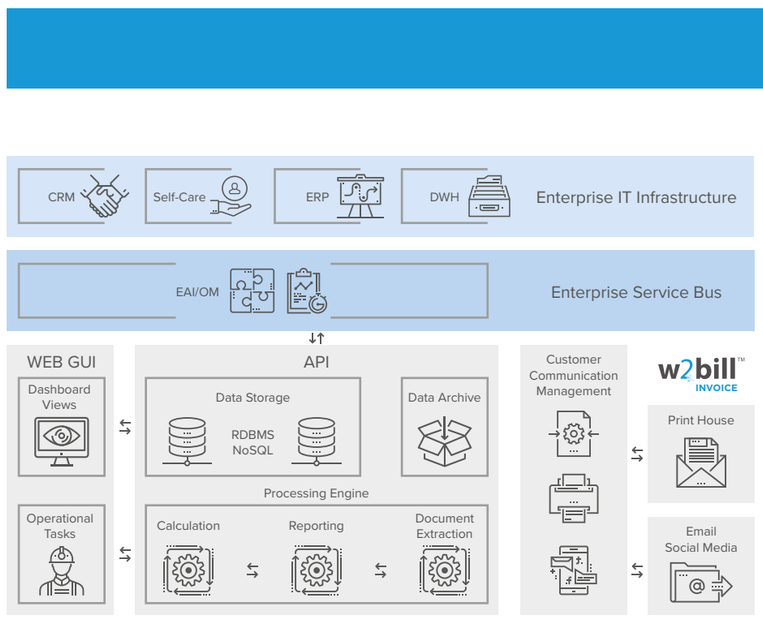

Figure 1. w2bill™ Architecture Overview

w2bill™ was designed, from its early stages, with the future in mind. Following old paths made little sense in a world marked by constant evolution, and with the growing certainty that today's solution will most likely be outdated soon. With such concerns in focus, different ways of achieving the set objectives were considered, where scalability, fault-tolerance, reliability, resilience and configurability were key. What is true today will soon be outdated and outpaced. The current acceptable amount of data will soon be surpassed. More processing power, in different geographies, through public clouds or private data centres, must be supported, with all the challenges therein considered and handled.

Core concepts behind w2bill™:

1. PERSISTENCY

It is immediately recognized that data must be kept safely, consistently and securely, to be properly and reliably used. Given the scope of the application involves billing and invoicing, it is paramount that no information is lost, that the data can be tracked and safe from unexpected manipulation. It must also scale easily, and have as few constraints in its management as possible.

Standard RDBMS's are tried-and-true platforms, that provide a multitude of capabilities most people take for granted but have shortcomings in either cost, scalability or performance. More recent NoSQL solutions have less wide-spread adoption in large enterprises, but commonly are inherently scalable and performant, at a fraction of the cost. And then there are ways closer to actual physical storage, like HDFS, ZFS, and others.

w2bill™ has selected a NoSQL solution to solve the persistency solution, with an added layer of security to ensure all interactions are handled properly. Nevertheless, it is designed to quickly adapt its operation to interact with standard RDBMS's, based on client needs or requirements.

With the inherent need of reliable and fast data access, it became evident retrieving existing information and storing new or modified one was best addressed with a specific solution, rather than rely on point-to-point access from the application to the databases. This approach also adhered to the philosophy of enabling rapid and seamless changes of underlying technology, which is core to the solutions implemented by CMAS.

For this purpose, a Data Grid solution was implemented to handle all aspects of actual data manipulation and access. This layer enables all interacting components to completely abstract on how and where the data is kept, they just access it in a performant and reliable way.

With this powerful capability, data integrity is assured, regardless of the actual persistency solution – or mix of solutions – chosen.

2. PROCESSING ENGINE

This is one of the most important components of the w2bill™ solution. It is a package of micro-services targeted at executing flows of tasks across platforms, carefully tailored for billing, invoicing and related needs, such as financial aggregations and reporting. Each of these flows has been configured for these main concerns but allowing further customization to address specific customer, market or regulatory needs.

Inherent to this engine is the capability of sequential or parallel processing, enabling the simultaneous handling of data for multiple purposes. Instead of following the traditional way of having discrete areas of action performing 'read-act-store' tasks, the engine performs the most possible actions in parallel. As an example, when processing one bill, at the same time the charges, discounts and taxes are being calculated, the necessary information for the configured reports is being managed as well. With this approach, when the report needs to be generated, only the remaining, missing data is effectively taken care of. The process will not have to read the raw information prepared by the billing actions to derive the information for reporting from scratch, wasting time and resources in this task.

For deployments having to deal with a large volume of bills and create recurrent reporting about distinct sets of bills, this approach effectively guarantees savings of time, whilst enabling real-time KPI's to be made available. As an example, if while processing 150.000 bills, it is necessary to understand how many products of a given category are being billed, it is possible through this approach to have them shown in the Frontend.

The features of the engine are accessible by the Frontend, which enables the inherent entities to be configured and accessed (Products, Customers, Invoices, Taxes, etc.), but it is also possible to integrate with other existing applications in any customer infrastructure.

This is achieved by leveraging the flows' configuration capabilities, allowing the access of external data in peripheral platforms. Such platforms can be responsible for providing sets of data, of varying levels of completeness, that are integrated into w2bill™ through a diversity of approaches. After dedicated transformation of the information into the w2bill™ own data model, the configured flows are executed similarly to any other ones. The integration itself can be simple – through pre-formatted flat files – or more complex – involving retrieving the data from an external database directly or through Web Services.

Regardless of the chosen option, w2bill™ can quickly be deployed in any infrastructure with minimum impact and allowing the customer to abstract from having to implement billing and invoicing related logic in platforms not suited for the purpose.

2.1 METRICS

All components are implemented to produce execution metrics and store them in a centralized repository where the system performance can be measured and monitored. With these metrics, it is possible to know how long each flow takes to be processed, as well as how long each component takes to execute the required operation. With these capabilities, it is possible to detect anomalies in compo-

nents' instances by monitoring the execution times of each in the system, along with other relevant information necessary for scheduling of performance enhancing actions, bottleneck detection or scaling decisions. Metrics are enriched with contextual data. It is possible to know the host where the component is running, the number of available cores and memory, which real-time group it belongs, along with additional relevant information. This is particularly relevant for large deployments having multiple machines, where the analysis can be further tracked.

2.2 LOGGING

In distributed deployments, component instances will be spread across several machines of a cluster. Each instance will produce logs that must be stored and grouped with the logs of all system components to allow analysis of the system behaviour. All components of w2bill™ are prepared to write application logs to a file – on the local machine –, and to send a logging message with additional data, allowing users to correlate all logs for a specific order or workflow execution. This enables a clearer big picture of the system. These logs are centralized in an Elasticsearch index with a predefined schema. With this approach and the built-in capabilities of Elasticsearch, it becomes possible to define analytics of the logged information.

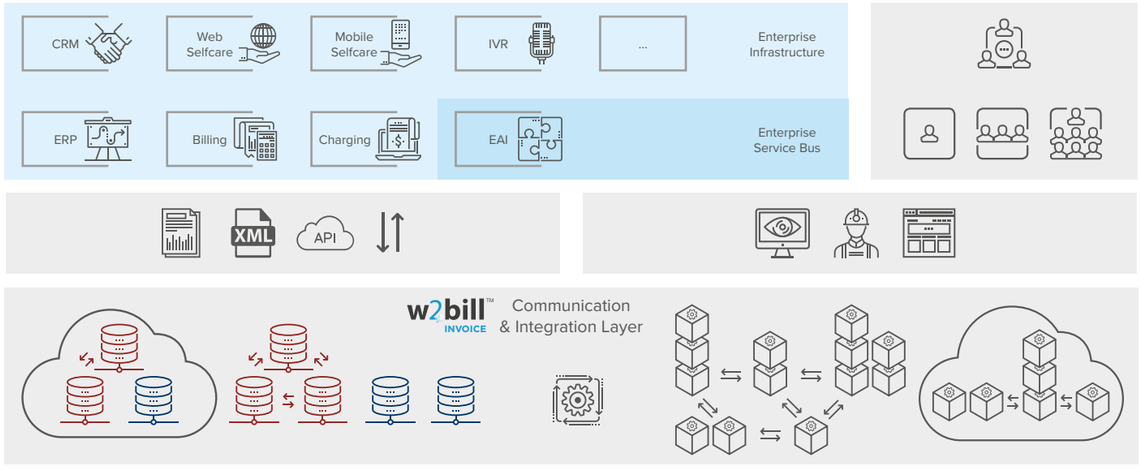

Figure 2. w2bill™ Integration Overview

2.3 FRONTEND

w2bill™ presents a GUI enabling common operations to be performed over the solution, such as interacting with its core entities under different profiles. There are three main profiles: System Administrator, Administrator and User, and each login is associated to one, granting the end user the respective privileges. The organization is then able to differentiate the people that use w2bill™ accordingly, to maximize its use and mitigate scenarios of improper operation.

The GUI allows the manipulation of entities such as accounts/customers, products, taxes, discounts, the creation and viewing of documents (invoices or others). Each user can execute actions only of the configured family of actions associated with its profile.

Even in scenarios where the application is used integrated into more complex infra-structures, the information available in the GUI allows the users to interact and with the data being processed. It this becomes possible to use the GUI for back office roles, enabling consultation, validation, creation and modification of the documents.

The Frontend itself is capable of being extended on a client-by-client basis, where additional data, or even layout can be added or modified to better suit the client's corporate look & feel or information needs. It is not intended as a static GUI with strict options and display, but as an adaptable layer to each of the customer's corporate image.

2.4 AUTHENTICATION SERVICE

To comply with the need of security and validated service access, w2bill™ embeds an authentication service enabling the services to be used only by properly authenticated users. With this feature, all requests made to the exposed services must be authenticated through a central authentication server – as provided by w2bill™ –, ensuring only the services each user has granted policies are accessed. This server implements the OAuth 2.0 protocol and can authenticate users in a relational database or a LDAP server.

2.5 TECHNOLOGIES

The entire solution is based on the technology stack presented below in more detail. In any case, given the architectural decisions made earlier on, it becomes fully possible to opt for specific solutions using different technologies, if certain key integration aspects are met. As an example, it is possible to deploy one component implemented over C/C++, or Microsoft C#, or even Python, to achieve certain necessary integration with other applications. Should it be necessary from a customer perspective, the choice in the persistency stack can be replaced by other NoSQL solutions such as MongoDB, or even more traditional RDBMS's like MySQL, Postgres or Oracle.

Across the entire solution, the choice of technology has been focused on its adequacy (for its given set of responsibilities), as well as its proven track-record, industry wide-spread adoption, reliable enterprise-grade support and evolution roadmap.

With the growing adoption of cloud-based approaches instead of on-premise, certain facets of architecture, design and underlying implementation were considered to assure the solution was prepared without need of specific customization. By addressing these concerns early, the selection of technologies was further refined, where only the ones with appropriate levels of compliance were left as eligible.

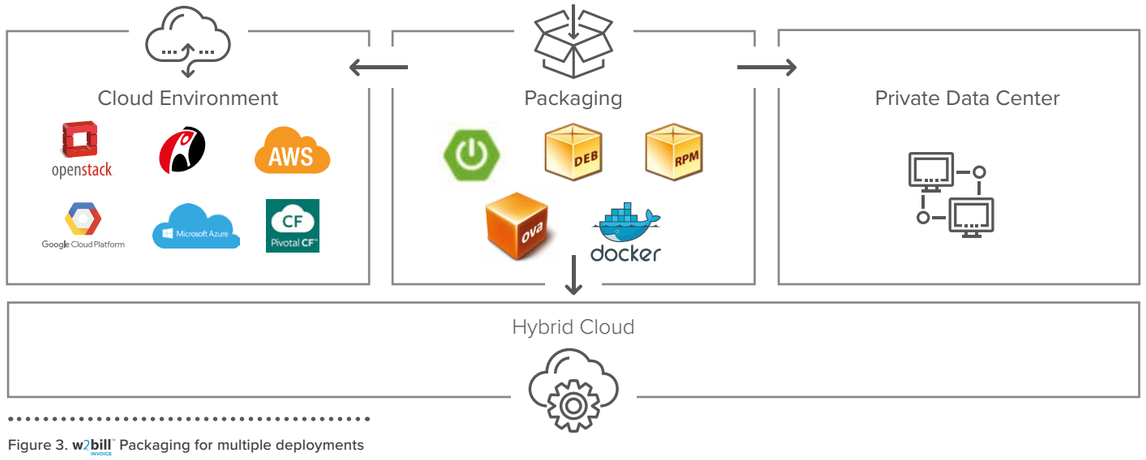

Figure 3. w2bill™ Packaging for multiple deployments

3. DATA ACCESSIBILITY

Consistency across all data is ensured. Knowing beforehand that a given customer may select different approaches w2bill™ has implemented a data grid solution to handle all aspects of data manipulation. Accessing this data grid is possible through multiple programming languages.

4. MULTIPLE CUSTOMER COMMUNICATION CHANNELS

To enable the communication and presentation of documents to the end customer, a series of innovative capabilities is introduced, allowing multiple output formats to be created. These formats can be of the more traditional sort, such as PDF files for wide-spread digital visualization or PS files to be sent directly to printers, but also innovative ones like dynamic HTML5. With the use of these dynamic documents, a new way of communication with the customer is enabled, presenting richly formatted information with the capability of bidirectional communication. This two-way action is done over the web, allowing partial document contents to be fetched when the customer opens the document – instead of being statically inserted in it –, or even provide feedback to the document issuer of when the document was opened, and which sections of it.

With this methodology, further analytics can be deployed by the issuer to better understand how its documents are handled by its own end clients, and with it improve its communication processes.

ROADMAP

Fully aware the product of today isn't the product of tomorrow since the needs and technology of today will surely be replaced in the future, the w2bill INVOICE solution has a constantly evolving roadmap of features. These new capabilities will come from the new experience gathered from our customers, as well as their insight, advice and suggestions.

There is no aim in having customer-specific branches, stopped in time over each deployment, but assure a recurring evolution targeted at providing new and better ways to support technology and business.

By constantly investigating on what is happening and about to happen, as well as converging experiences across markets, customers and mindsets, we envision the path of improvement to be made available. Partnership with our clients is key in the forming of this vision and the definition of the roadmap.

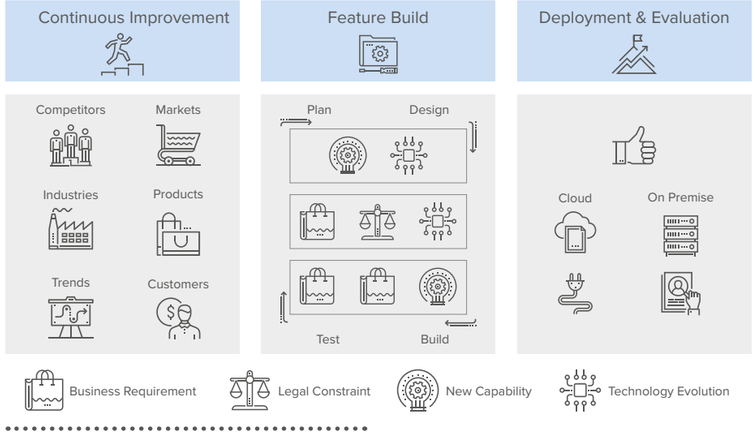

Figure 4. w2bill INVOICE Roadmap Approach

APPENDIX

Some technical insights on w2bill INVOICE Invoice

PERSISTENCY

APACHE CASSANDRA

The choice of Apache Cassandra for storage was based on its multitude of capabilities, particularly its decentralized nature, avoiding single points of failure or network-based bottlenecks. Its Fault Tolerance, through its data replication across nodes and data centres, together with the capability of failed node replacement without downtime. Also because of its scalability, by use of 'nearly infinite' multiple nodes to support processing and storage growth, its elasticity, by relying on its nodes increase for read/write throughput and the professional support and proven use on global enterprises (CERN, eBay, GitHub, Apple, Netflix to name a few).

w2bill INVOICE relies on Apache Cassandra solution to store its information across its components. The use is indirect, however, as the layer of accessibility is performed via Data Fabric Apache Ignite. It is the responsibility of this layer to effectively communicate with Apache Cassandra, for effective storage (write) and reading of data.

DATA ACCESSIBILITY

APACHE IGNITE

Apache Ignite is an In-memory Data Fabric that provides high-performance data, compute and service grids. It supports fully ACID-compliant distributed transactions, ensuring consistency across all data and supporting standard SQL syntax to query the objects stored in the data grid. Accessing this data grid is possible through multiple programming languages. Another relevant feature is the advanced clustering capabilities enabling scalability, fault-tolerance and high performance requirements.

Currently, w2bill INVOICE uses the data grid as a layer to interact with the persistency solution, with read-through or write-through approaches. The effective writing is executed in an asynchronous manner, to expedite performance.

PROCESSING ENGINE

JAVA 8

w2bill™ is built using the latest version of Java language. It enables to take advantage of all new features and performance improvements introduced with version 8. Java has been selected as the reference language given its wide industry adoption, the simplicity to deploy in multiple platforms, its code portability. Java has a widespread number of open source plugins and frameworks, an extremely active developer community, and an extensive evolution roadmap as well as a support group.

PIVOTAL SPRING FRAMEWORK

Pivotal Spring Framework is an open source framework that supports the development of Java applications, by providing help with infrastructure needs and supplying a consistent programming model over different technologies. It has been widely used throughout the Java development industry, as an alternative to the Enterprise Java Beans model.

Pivotal, and particularly its Spring team, are always planning the future and driving the framework to respond to new business requirements. Relevant examples are the Cloud Stream project and the introduction of the reactive programming in next release 5.0. w2bill™ will follow these evolutions closely to extract from them any relevant improvements.

PRESENTATION

ANGULAR JS

Angular JS is a widespread web application framework, targeted for front-end applications. It is an open-source solution, maintained by Google and a community of individuals and enterprises, aimed at Javascript-based development. If features a comprehensible set of tools enabling frontends to be developed for multiple platforms, taking into consideration its characteristics.

RESOURCE MANAGEMENT

APACHE MESOS

w2bill™ is a naturally distributed system. Any of its components and respective instances can be executed in multiple machines, thus contributing for better performance through horizontal scalability, as well as avoiding central points of failure, by relying on clustering and fault-tolerance techniques.

The implementation is designed to use not only common virtualization of machines but also containerization approaches, such as Docker or Kubernetes.

The distribution and resource management are thus a concern that has been properly addressed using Apache Mesos, which features centralized handling for deployment and scaling of w2bill™ 's components in any sort of installation, from on premise to cloud or a mix of both.

CMAS
SYSTEMS CONSULTANTS
Av. Dom João II, nº 44 C, 2.2 • 1990–095 Lisboa, Portugal
T: +351 919531710 • mail@cmas-systems.com • www.cmas-systems.com

CMAS – Systems Consultants, Lda
Av. Dom João II, nº 44 C, 2.2
1990-095 Lisboa
T: +351 919531710
mail@cmas-systems.com
www.cmas-systems.com