The Infor Architecture
Table of Contents

3 The inflection point
4 The Infor Architecture: A new approach
5 Infor ION: An Internet Architecture
6 ION @ Work: An example
7 How Infor ION works
   Step 1: XML business documents (BODs)
   Step 2: Connect to the service bus
   Step 3: Service bus routes the document
12 Infor ION—a complete platform
   1. ION Connect
   2. ION Event Management
   3. ION Workflow
   4. ION Pulse API
   5. ION Grid
   6. ION Desk
   7. ION Registry
   8. ION in the cloud
16 Infor Ming.le™: Social collaboration
   Capturing all enterprise activities
   Contextual applications
   Social Business Graph
   Structured collaboration
   Communities
   Infor federated security
20 About Infor
The inflection point

Systems integrators generate over $300 billion in annual revenue by implementing and integrating enterprise applications. Even a casual observer would accept that as compelling evidence the business software industry needs a radically simpler and different approach to orchestrating and integrating business processes.

After decades of shifting standards, interfaces, and a plethora of byzantine middleware suites, integration between applications remains labor intensive, fragile, and an impediment to innovation, rather than a facilitator of it.

It's time for a change.

Infor® had the opportunity, resources, and incentive to completely rethink enterprise architecture and leverage de facto Internet standards that have evolved—more modern and simpler technologies.

Infor also accepted and embraced the reality of heterogeneous application environments instead of pretending all business problems for all industries and functions can be solved by a single, monolithic, bloated piece of software.

The monolithic application suite era has ended, partly because all application vendors have made hundreds of acquisitions and no longer have a single suite; they’ve created integration challenges within their own product lines. The industry is moving toward a new era of loosely coupled applications based on standards.

The monolithic uber-suite is not realistic, given the vast differences between industries and microverticals and specialized processes that are mission critical. A general ledger is not a complete business suite. Requiring all applications to share a common data model to be truly integrated is a false hope. Modern computing architectures assume application diversity, and they design for that reality with simple semantic standards that can map to any data model.

The industry has evolved, but legacy middleware has not. Every other layer of the computing stack has adopted recognized standards, but communication between business applications is still a proprietary quagmire.
Legacy middleware had several issues:

1. **Transformation bottleneck:** A central hub, which transformed data and transactions between all applications, attempted to understand all application message formats. This complex and constantly expanding canonical model could not keep up with the velocity of change in application data models and APIs.

2. **Protocol complexity:** Vendors exposed thousands of granular web services without business context. Instead of sending a sales order, developers were burdened with hundreds of service calls for each field on the sales order, and each could be called independently. The analogy would be reading a book by getting a single page mailed to you daily.

3. **Fragile integrations:** Point-to-point integrations broke during upgrades because web services, application interfaces, and data models changed with each new release. Legacy middleware presumed a static model that was inconsistent with innovation.

We now know that attempting to make a single piece of middleware understand millions of intricate data structures and service call interfaces to every application ever built doesn’t scale and is impractical.

Enormous middleware suites have become an industry unto themselves, but are now stagnating under the weight of the complexity they perpetuate. The plethora of edge conditions they attempt to address is not relevant for the majority of business integration scenarios, but middleware bloat continues to expand. It’s time for a new direction.

**The Infor Architecture: A new approach**

As an applications-focused company, Infor's interest is aligned with customers; our goal is to radically simplify integration, minimize infrastructure investments, and think beyond integration to a scalable, yet simple information architecture that enables collaboration and normalizes big data.

Infor adopted five architectural principles modeled on the Internet:

1. Applications should publish all events in real time in standard, complete XML documents defined using a standard core component architecture.

2. Subscribing applications should be unaware of and fully insulated from upgrades and other changes to publishing applications.

3. This publish and subscribe architecture moves transformation tasks to the publishing applications instead of tasking a single piece of middleware with understanding thousands of APIs and message formats across all applications.

4. All new or updating transactions should be communicated and retained as complete, self-describing XML documents with all fields and metadata to enable big data.

5. All published events should be captured in real time and preserved, creating a big data repository for business intelligence without batch loads and extracts.
The Infor Architecture consists of:

- **Infor ION**—Standards-based middleware that unifies business processes by routing XML documents between applications using a publish and subscribe model. It also retains copies of those documents in a big data central repository (Business Vault).

- **Infor Ming.le™**—A social user experience built on ION that unifies structured processes in applications with unstructured collaboration.

### Infor ION: An Internet architecture

At the core of the architecture, Infor wants to run business processes in a distributed and loosely coupled way. We call it the “Architecture of the Internet.”

The Internet does not have one main web server to host all web sites of the world; it has millions of web servers, and it has no single piece of middleware orchestrating all conversations. Likewise, the Infor Architecture has no central hub (bottleneck) transforming all messages and processes. Applications simply publish all events and subscribe to relevant events in other applications, much like an RSS feed.

A fundamental objective in modern middleware is to eliminate tightly coupled, static integrations that often break as applications change or during upgrades. As a simple analogy, traditional integration is more like email. To communicate, you need to know the specific address of the target; and if that address changes, the communication will fail. ION is more like Twitter® in that you can publish as much as you like, and any consumer interested can subscribe to that feed.

### Traditional middleware vs. Infor ION

<table>
<thead>
<tr>
<th>Analogy</th>
<th>Traditional middleware</th>
<th>Infor ION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Point-to-point</td>
<td>Publish and subscribe</td>
</tr>
<tr>
<td>Source application</td>
<td>Called by specific application</td>
<td>Publishes to all</td>
</tr>
<tr>
<td>Transformation</td>
<td>Middleware transforms all messages</td>
<td>Each app publishes XML</td>
</tr>
<tr>
<td>Implementation</td>
<td>SOA service calls to specific fields</td>
<td>Complete, standard XML business documents</td>
</tr>
<tr>
<td>Standards</td>
<td>Thousands of proprietary APIs</td>
<td>Standards-based complete business documents</td>
</tr>
<tr>
<td>Complexity</td>
<td>80 disks</td>
<td>3 disks</td>
</tr>
<tr>
<td>Resilience</td>
<td>Breaks on upgrades</td>
<td>Protected on upgrades</td>
</tr>
<tr>
<td>Big Data</td>
<td>Dated warehouse</td>
<td>Real-time Business Vault</td>
</tr>
</tbody>
</table>

Infor is in no way committing to the development or delivery of any specified enhancement, upgrade, product or functionality. See “disclaimer” paragraph contained herein.
ION @ Work: An example

Communicating a new order from the source order entry application to all other applications that need to know about that order is a typical use case for ION.

The creation of an order in an ERP application would trigger the creation of an XML document, which is put on the ION service bus. The service bus then routes that sales order to all subscribing applications interested in the order, such as warehouse management, supply chain, or a third-party app.

The order entry application, which is the publisher in this example, converts the order to XML format and eliminates the need for the middleware to transform the message to some alternative, proprietary format. The publishing application has the most knowledge of its own data model and is in the best position to do the conversion.

The subscribing application knows the sales order is coming in the same standard XML format, no matter which application published it. If the publishing application changes its data model or APIs, the subscribing warehouse application is not affected, since it only needs to get the same standard XML document it always receives.

The sales order would look the same regardless of whether it came from an ERP system, a call center application, Infor’s Rhythm Ecommerce service, or a third-party partner application. A sales order is the same XML format in every context, which simplifies communication. Instead of each application talking a different language, Infor uses industry-standard XML documents based on UN/CEFACT core components (e.g. OAGiS) for all applications. It’s analogous to the enlightening moment when a country decides to speak one language for the sake of efficiency.
A copy of the XML document is also routed to a central repository called the Business Vault. The Business Vault, a real-time data mart, is the underpinning of real-time big data that can be used for business intelligence and is one more subscriber to the document. The Business Vault is a universal repository with many use cases in the Infor Architecture. The data is available in real time for in-context business intelligence in the presentation layer.

Other subscribing applications, whether in the cloud or on-premises, get the same sales order in the same XML format. A subscribing application could be a third-party application like Salesforce.com, a mobile application, or a sales dashboard.

How Infor ION works

At a summary level, there are three major steps to move information between applications in the Infor architecture:

---

**Step 1:** Create XML Business Object Document (BOD).

**Step 2:** Transfer the BOD to the Service Bus.

**Step 3:** The service bus routes the BOD to all subscribing applications and sends a copy to the Business Vault.

---

**Step 1: XML business documents (BODs)**

At Infor, we have enabled all major applications to natively publish or subscribe to XML documents in our standards-based format. We call them business object documents (BODs). Instead of defining a new proprietary format, Infor has worked with standards organizations to adopt a standard architecture, while maintaining the flexibility to use market specific vocabularies.

The business object document is a clear and unique business representation of a business object (e.g., invoice, customer, asset, order) with complete metadata. A BOD is self-describing and standard. There is no need to have ten different BODs for the sales order; Infor has only one. So the same sales order is produced by an ERP, CRM, etc. and received by a warehouse management system or financial application.
Adopting the standard architecture for all applications allows all Infor and non-Infor applications to share common semantics. There is no mapping required in the service bus. When configuring a business process, the analyst would focus on the business routing rules without worrying about how certain fields would map into each other.

Having a BOD produced with all the information you need has major benefits in the deployment architecture:

- **No query back to fetch additional data.** We have seen many corporations having different touch points to retrieve a different data set for the same business entity. For example, a large enterprise had 121 touch points to retrieve a different query result of the sales order table. Imagine, you can remove 121 touch points to a single, backward-compatible touch point;

- **The introduction of a central information repository,** called Business Vault, where all data that came through BODs can be automatically stored and ready to be used for BI and operational reporting. Business Vault is just one of the subscribers, but would receive all the events and store the information in a schema that can be utilized by BI or operational reporting systems. Since this happens on events, you don’t need to run complex synchronization scripts during the night; it happens automatically as business runs. This again removes a number of touch points back to your business system;

- **Real-time activity monitoring for events based on business rules.** Typically, business monitoring rules are application specific, which means they require the definition and maintenance of different and potentially conflicting rules for the same exception. For example, you may monitor opportunities that are lost above $100,000. If you have multiple CRM or ERP systems, you will need to monitor each of them and set up application-specific rules, as their data models are different. At Infor, you define one single business rule using the elements from the standard XML document. So the rule is defined once, and the source of the document is irrelevant. Since it’s a push behavior, the monitors do not need to query back to your transactional database every minute. This would again remove a number of touch points back to your business system;

- **Workflow approvals and authorizations based on business rules.** Normally, you would deploy an additional workflow system to run your approvals, authorizations, etc. This would introduce another set of different touch points. At Infor, the publication of an unapproved requisition can be set up to be subjected to a workflow approval, regardless of the source system. The workflow rules are defined using standard XML elements, like in the business monitoring rules example.

- **Sending to or receiving BODs from new (mobile) applications in real time or disconnected.** You can build a new (mobile) order entry or order status application, and the information can be federated in one single view, regardless of whether it comes from different systems;
This architectural approach dramatically reduces the length and costs of an implementation. Infor has made the necessary investments to architecturally enable our applications to support BODs and guarantee their backward compatibility to support upgrades. This is unprecedented in the software industry, as most software vendors have failed to execute this type of strategy.

**BOD metadata**

When a BOD is created and transmitted, it carries next to the business data an envelope with key, core information to identify itself.

1. **Tenant ID**

   This ID represents the tenant to which this information belongs. This enablement is right at the core of each and every BOD and marks the enablement of hybrid or cloud-only deployments. Additionally, this represents the financial company ID where the transactions belong. This is critical for applying security, and routing and filtering rules.

2. **Accounting Entity ID**

   This ID represents a financial level within the company where the transactions belong. This is critical for applying security, and routing and filtering rules.

3. **Location ID**

   This ID represents the physical location that is associated with a transaction, like an office for a sales order or a warehouse for a shipment.

4. **Version ID**

   This ID represents a version of a sales order, etc. This is critical when, for instance, you want to monitor for changes, such as someone modifying or deleting lines in a sales orders, or keep all historical data to be used in the business vault.

5. **Logical ID**

   A logical ID is a unique identifier for an application. This is constructed with a prefix for Infor applications, like infor.eam, followed by the ID of the physical instance. In essence, wherever a BOD is or travels in the Infor network, it can always be tracked, and its origin can be always identified. This ID has another major advantage: It helps to construct the URL to drill back, like navigating from a reporting application into the ERP application where that sales order is maintained.

For pieces of information that are not defined by the standard definition, Infor has created a registry where customers can create their own BODs and store them. There are known as custom BODs.

BODs like a sales order, invoice, shipment, etc. contain all the information you need to power any business process integration without the need to query back. BODs are an independent, self-sustainable unit of information; but they also carry foreign keys of related information, like the sales order key in a shipment BOD or a requisition key in a purchase order BOD after its conversion.
BOD references are powerful because they enable the following functionalities:

- **Business Process Monitoring**, where the spotlight is not on monitoring a single BOD entity but on a business process like requisition, purchase order shipment, invoice, etc.

- **Social Business Graph**, which gives a unified overview of an entity, like a requisition, and all related conversations, alerts, tasks, and notifications in one view. In addition, the graph gives you, in real time, the progress of a business process; as soon as the requisition is converted into a purchase order, the graph will connect that purchase order to its original requisitions because of the BOD reference (remember that the requisition and the purchase order can be in different systems). Then all conversations, alerts, approvals, etc. can be seen around that purchase order as well. Graph is a powerful visualization of a structured business process augmented with unstructured information, and it represents the next generation of enterprise search.

- **BI and Operational Reporting**, where BODs can be routed to the business vault. When BODs get shredded into a purpose-built schema, they use the references to populate the data warehouse.

**Security around BODs**

Not everybody can follow any information, as this would create a security issue. That is why a BOD carries metadata, so it can be filtered by roles, accounting entity and location, and even by BOD type/category, like financial, HR, etc. When a BOD is followed, not all information is disclosed in the feed. Mostly, it’s only the document ID and the status that changed (like from New to Approved). For additional information, a URL for drill back is given to end users to navigate and view data in the application, where all authorizations are applied as well.

**Step 2: Connect to the service bus**

The most trusted way to connect an application to a service bus in an asynchronous way is to use, for example, Java Messaging System (JMS).

For applications that do not support JMS, Infor delivers a mechanism that all applications have in common, either modern or traditional, i.e. a connection to a database. If an application can publish or receive an XML document via a database, then it can also be plugged into the Infor Architecture. This mechanism is called the I/O Box. Almost any application can exchange XML documents.

The I/O Box functions like a queue, and it is processed according to the business routing rules defined in the system. When an application produces an XML document, it puts the document in its outbox table. The service bus picks up the XML document and routes it to the inbox tables of the subscriber. Subscribers will be notified and start to process the documents.
Important, the service bus does not need to perform a heavy-duty transformation step at all. When the XML document reaches the subscriber’s inbox, the Infor application has been enabled to process it natively.

XML documents are queued if a subscribing application is down and sent when the application comes back online. This would not affect other subscribers, and (sub) processes can continue to be executed. If the service bus is offline, processes can continue the execution; ION will pick up the messages from the outbox once the service comes back online.

ION offers prepackaged connectors to Infor, Oracle®, and SAP applications, as well as others. In addition, ION has been certified to connect to WebSphere®, Tibco, and Oracle middleware.

**Step 3: Service bus routes the document**

Infor has selected a lightweight queuing technology, Apache Active MQ (service bus) to run all the routings of messages between applications and for reliable transmission.

Active MQ is a proven open source product used by Google®, Amazon®, and many other large companies, so the underlying service bus for ION is a robust pipe. Infor has defined what needs to run over those pipes for business applications and added our own workflow and data mapping.

All routing instructions are created and executed centrally but, like the Internet, executed locally.

The service bus delivers the BOD to all subscribing applications. A subscribing application receives the BOD with full metadata and can act on the content it cares about. Each change in a business document, e.g., a changed sales order, triggers the production of a BOD, and the full document is published. This creates a full audit trail with no missing data.

If third-party applications cannot process Infor BODs natively, then ION features two mapping engines: an XSLT mapper and also an internal mapping tool based on Eclipse, which generates Java transformation code.

The service bus also sends a copy of the BOD to the Infor Business Vault. The Business Vault is a real-time data mart that receives all iterations of a BOD. If a BOD is changed seven times, there are seven copies in the Business Vault.

Business Vault is real time in the sense that it requires no batch loads or ETL (extract, transformation, load) processes.
Infor ION—A complete platform

Infor ION is a single platform that delivers integrated services and tools out of the box that allow for complete business process orchestration:

1. ION Connect

ION Connect is a collection of prepackaged connectors to business applications and third-party middleware products.

Infor applications will use the standard I/O Box connector as discussed previously, but third-party applications could use it as well or utilize one of the additional connectors, like web services, generic database, file, etc.

For integrations with third-party middleware, ION can integrate through JMS (both as client and server mode). It has been certified with IBM WebSphere, Oracle Fusion, WebMethods, Tibco etc., enabling a corporation to connect the Infor application landscape to ION, take advantage of all the enablement, and then connect ION to an existing ecosystem that the organization has put in place.

Additionally, ION offers prepackaged business adapters, like SAP, Oracle, and SalesForce.com. These adapters would not only provide the connection to the business application, but also a set of prepackaged mappings between the standard Infor canonical model and the proprietary application format.

ION Connect uses protocols like HTTPS, secure FTP, etc. to ensure data is transmitted in an encrypted and secure way.

ION Connect also prepackages the Infor Cloud Connector, enabling BODs to be sent to or received from Infor applications that are deployed in Infor CloudSuite. For example, you may run ERP in your IT infrastructure, but you may have decided to run Enterprise Asset Management and/or Expense Management in Infor CloudSuite.

ION is indifferent to whether an application is running on-premises or in the cloud. From an ION standpoint, it is just a connection reconfiguration (now it would point to Infor CloudSuite), re-activate and go. During that process, all BODs are queued and ready to be processed later in the cloud.

If third-party applications cannot process Infor’s standard BODs, then they can use one of ION Connect’s two additional mapping engines: an XSLT mapper and also an internal mapping tool based on Eclipse, which generates Java transformation code.

Finally, Infor deploys asynchronous, event-driven business processes by default, but there are some use cases where a traditional request/response is required, for example, in applications like e-commerce where price or available to promised are requested. ION Connect features web services or can expose itself as a web service with both a SOAP and REST interface.
2. ION Event Management

As BODs are transported from the publishing application to the subscribing applications, there is much more you can do with that information. For example, in ION Event Management, you can define and activate monitoring rules. The setup can be done by a business analyst, as programming is not required.

When a rule is activated, ION Event Management listens to the interested BODs, applies the rules, and sends real-time alerts to people or to a group of people that an exception occurred. Additionally, next to a single event, ION Event Management can also monitor a business process.

By using the relationship metadata built into the BODs, ION can relate a sales order originated from an ERP to a shipment that was originated by a warehouse management system. This allows users to apply rules to different events so they can find out that their shipment is late for that important customer.

Finally, ION Event Management can detect non-events—things that were supposed to happen but did not. What happens today if you did not ship an order for an important customer? What if you haven’t received a payment? Since ION Event Management is time based as well, it will notify a subscriber when, for example, there has been no payment event for an invoice by the due date in the original BOD.

Next to saving all the BODs for BI and reporting purposes in the Business Vault, the exceptions and monitors can be stored as well, unlocking BI for business process management.

3. ION Workflow

ION Workflow utilizes the same business rules engine used in Event Management to detect if a workflow approval needs to be executed. We call this activation policies. Workflow is defined in the ION Desk using a graphical editor that is based on BPMN 2.0 standards.

Standard patterns like routing, conditions, loop-backs, matrices, etc. are supported. When a workflow runs, different tasks or notifications are sent to people or a group of people. A task can be approving a purchase order, or reviewing a new price, or a link to an application to enter or review some information.
An ION Workflow can be activated in different ways:

- **Through a BOD event** (transactional or master data), like new requisition, new item, etc.
- **From an alert**. Alerts notify users of an exception, but Workflow could also help resolve the exception using a standard procedure.
- **Through a Workflow BOD natively**, where an application uses ION Workflow as an embedded engine. In this case, in the Workflow BOD, there are the name of the workflow, the input parameters, and the expected output parameters.
- **Manually**, by uploading a PDF or other format and routing that document through approval. This would use the same workflow approval as in the first method, with the difference being that the input is not a BOD but a document.

4. **ION Pulse API**

ION Pulse is the architectural layer on top of ION that collects all the business events, alerts, tasks, and notifications, exposing them as a set of APIs to be used by applications like Infor Ming.le to push all the work in real time to end users. The ION Pulse API implements asynchronous notification patterns. Whenever there are new items, ION Pulse notifies the applications to fetch them.

5. **ION Grid**

The ION Grid is a pluggable platform that provides intelligent scaling, load balancing, fail-over, and self-healing capabilities to Java-based services, in this case the ION services. The ION Grid runs on top of the Java Virtual Machine (JVM) and allows distributed programming without concern for the physical location of the service. It embeds a J2EE servlet container to run web applications. For example, the ION Desk application utilizes it.

The advanced scaling provided by ION Grid has been uniquely designed in ION. Every step output in the business process is given back to the grid queue: the grid then determines which host is less busy to execute the next step. So, the grid scales not only a business process, but also the steps inside it.

Another feature is what we call self-healing. When a JVM crashes because of memory issues or ION itself has a problem, the ION Grid is capable of restarting the services automatically and continuing the processing where it left off. This process works automatically and does not need system administrators to intervene. All activities are logged and also sent to the administrators as email.
6. ION Desk

The ION Desk is the one single place to configure and administer all ION Services by tenant.

It starts by registering the applications (Infor or non-Infor) and configuring which BODs can be handled by which application. This information can be imported or exported. Once registered, business integration models can be created or imported as well. For example, you can model a system-to-system flow (we call it document flow), business monitors, workflows, service flows for SOAP or REST type of integrations, mappings, etc.

When all models are created and validated, they can then be activated centrally. From a deployment perspective, ION Grid allows for the running of multiple hosts with different ION Services available. When ION Desk activates the models, it connects to the ION Grid, which will propagate the configurations across the different nodes and hosts. This is modeled centrally, and execution is distributed.

Further, in ION Desk, you can administer the complete operations of the ION service. Within ION Desk, there is a module called OneView that features a BOD or Service Timeline diagram. Instead of searching through text-based log files, OneView gives a graphical overview of what happened in ION when a BOD event occurred. This gives system administrators consumer-based logging, which makes it easy to understand the operations and quickly find issues and resolve them.

7. ION Registry

The ION Registry holds all XML schemas of the BODs used by Infor applications. Customers can also register their own custom BODs, and use them in ION the same way they would use standard ones. A customer can also add fields to the standard BODs, but this is allowed only in the user area segment of the BOD, to avoid breaking the backward compatibility when upgrading.

So, if the application has been modified (for example, a couple of fields in the sales order table have been added), these fields can be made available in the user area of the sales order BOD. These fields are then available for Business Vault, Event Management, Workflow, etc. Today, the ION Registry holds more than 100 BOD definitions.

8. ION in the cloud

ION is a cloud-ready middleware suite. ION can route BODs between cloud applications and also run in the cloud itself. It is multi-tenant enabled. ION can also run in the cloud but integrate two on-premise applications and any hybrid permutation of cloud and on-premise deployment.

The Business Vault can run in the cloud or on premise. It can also be federated, which means multiple Business Vaults configured by division or geography, all running in the cloud, on-premises, or a hybrid of both.
Infor Ming.le: Social collaboration

While Infor ION takes care of all back-end business processes, harmonizes the information, and pushes work through events to end-users, Infor Ming.le takes care of all front-end business processes and the presentation layer.

Infor Ming.le is a social portal to all applications, alerts, tasks, and business intelligence—an enterprise news feed integrated into all applications, data, assets, and employees. Infor Ming.le brings social collaboration, analytics, and business process management into one single platform to end users. All business events generated and pushed by ION will manifest to end users through Infor Ming.le on a desktop or mobile device.

When a user logs into Infor Ming.le, a home page is displayed. It provides two different modes to work with; a user can select the one they prefer. There’s a traditional cockpit view, where all work (alerts, tasks, etc.), including business intelligence, for that specific user is displayed; or a conversational mode, like a feed in social media websites, where all work is also presented by time, but mingled with conversations around a specific business process, issue, or topic. This intentional dual mode is provided to support the new generation of people entering the enterprise world, who are more comfortable working in this way.

Capturing all enterprise activities

One of the key goals for Infor Ming.le, aside from providing a canvas for applications to run within and produce and manage context, is to capture all enterprise conversations around business processes. When you buy an ERP or CRM application, you are interested in structuring certain business processes, like purchasing, sales, manufacturing, etc.
In reality, this covers only a percentage of your enterprise business processes or activities. Whenever there is an issue, such as a rush order from an important customer or a purchase order that needs to be approved, employees immediately use other tools to find people for help. So people start making telephone calls, using chat programs, sending emails, etc. This valuable information is never captured; it sits in people’s inboxes and on their laptops. When people leave the organization, it will be gone forever.

This is intellectual property that organizations are not capturing at all. Infor Ming.le allows you to find and connect to an expert and share or request information in context of an alert or a task you need to perform. The conversations, documents, and resolutions are all captured in one single place. It is searchable and can be found later as the audit trail of a decision process.

Aside from working in the Infor Ming.le application, a user can access applications registered in Infor Ming.le, based on the roles and authorizations he or she has been assigned. The user can automatically access one of these applications, thanks to the single-sign-on capability.

A key benefit of ION’s business document approach via XML is that those same BODs are available in real time for event management, alerting, and workflow. ION inspects those BODs and fires off alerts to subscribing users. So, not only can applications subscribe to a BOD, people can also. A user can decide to “follow” a sales order, and every event on that sales order will trigger an alert in their Infor Ming.le feed.

Contextual applications

Infor Ming.le defines the boundaries and the layout of how Infor applications are presented to end users. It offers an entry point, like a portal, to Infor and non-Infor applications.

The canvas has been designed so that each application screen is accompanied by a set of contextual applications on the right side. When you are in a sales order screen, entering an order for a customer, contextual applications that appear automatically on the right side of the screen could show inventory by item (in-context BI), or a map for the shipping address, or alerts for that specific customer.
The contextual applications are reaching into the Business Vault and extracting information from BODs in real time. This is human-to-human collaboration.

Infor’s vision is to go beyond that and give end users the capability to easily subscribe to (let’s call it follow) information, and get notified every time there is an update. This is system-to-people collaboration. Once a BOD is followed, it can be tagged, and there can be conversations, replies, etc. It then becomes a social object.

The need to have immediate access to information that relates to the task a user is performing is also very common in consumer websites. Think about what happens when you order something from a website—additional or background information is offered to you to make your decision faster and better.

Just as Infor enabled applications to publish or receive BODs through Infor ION, we have enabled applications to publish or receive contextual events within certain screens. A contextual event (we call it Infor Business Context), formatted in JSON standard, is published by the application screen with relevant information inside that specific screen.

The outlined area to the right of the screen below is reserved for Context Apps. Those Context Apps provide real-time intelligence relevant to the transaction to the left of the screen.

For example, in a sales order screen, the JSON message would include the sales order id, customer id, item id, shipping address, etc. As with BODs, the JSON message is enriched with header information like the tenant ID, the accounting entity, location, and document type. This allows security to be applied by role within a certain tenant, etc.

Armed with this data, the Context App can render anything in the Business Vault that is relevant to the transaction, e.g., last five purchases by that customer, inventory on hand for that SKU on the order, the sales rep assigned, etc. This is a major benefit of Business Vault and the Infor Architecture.
This is a big data architecture that makes all relevant data available all the time, in context. The customer can decide what to render in the context pane.

To enable the Infor Business Context, applications need to add a JavaScript library to start publishing or receiving these messages. The library is part of the Infor Ming.le SDK.

Social Business Graph

Social collaboration doesn’t happen only around an issue or a task; it could start by having people following or subscribing to information. As described earlier, we have enabled collaboration around every BOD type, as well. This collaboration starts when end-users initiate a “follow” in Infor Ming.le (if, and only if, the security setup allows it). Users can right-click during any process and decide to follow an order, invoice, customer, or even an asset. From that point forward, changes to that object trigger an alert to their Infor Ming.le feed.

Follow Objects

Infor Ming.le is so powerful because it is platform agnostic when it comes to the business domain an object is part of. Other software vendors offer collaboration around a certain domain, for example CRM. With more than 100 industry-standard BODs defined, information can propagate to all kind of domains, and organizations can collaborate around sales, purchasing, finance, HR, manufacturing, etc. The platform is decoupled from the content.

As people collaborate around a certain issue or a social object, a URL is always available to drill back to the original application that created that transaction. By using the security setup and the federated SSO, end users can navigate immediately to the application to discover more information. URL discovery and presentation are done seamlessly from the end user’s point of view.
Structured collaboration

The feed is not always the best way to structure collaboration around a certain topic or with just a group of people. In reality, since everything is time based, topics can be mixed throughout the feed. In Infor Ming.le, you can also structure an enterprise initiative around a stream, like the launch of a new product, a marketing campaign, a new hire, etc.

The stream has an owner, invites a select number of people, and will determine the topic of the discussion. A stream can be used for conversations, but also for making decisions, like approving a new brochure, new pricing for a new product, etc. Manual or on-demand workflows can be initiated as well, relaying them on the ION Workflow engine.

Communities

External users, like distributors, dealers, or partners can be invited to a certain stream and be part of a decision-making process, participate in a poll, or be updated on an upcoming new product launch. External users can log in into Infor Ming.le by registering their identity with their own email in the Infor cloud service. Just as you can log in to some websites using your Facebook account, your Infor identity can be used to sign in to the Infor Ming.le application.
Infor federated security

In the consumer world, single-sign-on (SSO) is well recognized. Once authenticated by a Google or Facebook account, a consumer can sign on to different websites. The question is: Why is SSO not yet possible in the enterprise application world?

It is possible today, but the issue is that enterprise applications were too late to agree on and adopt standards. At Infor, we made a huge investment in changing the authentication of our applications to adhere to standards and deliver true SSO capabilities. Within Infor Ming.le, we use the following standards to authenticate users:

- Adoption of federated technologies like Microsoft® Active Directory Federation Services (ADFS), using claim-based authentication processing user information in a trusted token. Applications can use SAML 2.0 or WS-* to integrate to ADFS-compliant systems. By using these standards, Infor applications and Infor Ming.le can integrate with other federation technologies, as well.

- Integration with Microsoft Active Directory for username and password authentication. Instead of relying on proprietary user stores for each application, users can sign on with their corporate account, inherit their groups, manage hierarchy (if implemented), etc.

- Infor Federation Services, which features an attribute store to register, applications (with URLs, etc.), roles, additional user attributes, distribution lists, groups, contacts, authorizations for social objects and for workflows, etc.

About Infor

Infor is the third largest business applications company in the world with 70,000 customers, 13,000 employees, and 4 million subscribers in the cloud.

Headquartered in New York City, Infor creates beautiful software that is purpose built for specific industries.
About Infor

Infor is fundamentally changing the way information is published and consumed in the enterprise, helping 70,000 customers in 200 plus countries improve operations, drive growth, and quickly adapt to changes in business demands. To learn more about Infor, please visit www.infor.com.

Disclaimer

This document reflects the direction Infor may take with regard to the specific product(s) described in this document, all of which is subject to change by Infor in its sole discretion, with or without notice to you. This document is not a commitment to you in any way and you should not rely on this document or any of its content in making any decision. Infor is not committing to develop or deliver any specified enhancement, upgrade, product or functionality, even if such is described in this document.

Copyright © 2014 Infor. All rights reserved. The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All other trademarks listed herein are the property of their respective owners. This document is provided for informational purposes only and does not constitute a commitment to you in any way. The information, products and services described herein are subject to change at any time without notice. www.infor.com.