

# OFBiz in the fashion industry, an omnichannel approach

ERP in SMEs: The Apache way!

Nicola Mazzone  
nicola.mazzone@mpstyle.it

- Who I am? What I do?
- ERP in SMEs
- OFBiz...and the third way
- Case History
- Omnichannel
- Q&A

- **Nicola Mazzone**
  - Now Ceo/Cto of Mp Style
  - Project Manager / Developer since 2000 on ERP & e-commerce Projects (in the fashion Industry)
  - Skills
    - Managing small development group
    - Java Programming
    - IBM mainframe
    - Process management in clothing manufacturing companies
    - Omnichannel strategy
  - Enthusiastic about Open Source & OFBiz!

**Active!** Every  
day currently  
maintained  
and developed



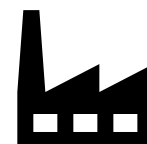
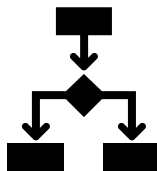
Run on IBM  
**Mainframe**



**Vertical Solution.**  
ERP only for fashion  
industry

## MpStyle ERP

It is a "**standard**"  
ERP, in practice all  
customers have the  
same database and  
the same procedures

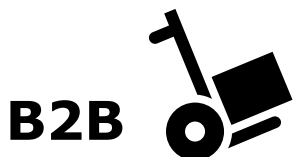


We have dozens of  
installations. Our target is the  
**small / medium company.**  
Our typical customer has 20  
to 150 users

By using the skills of the processes in fashion, to offer our historical customers and new customers the 2.0 version of what is today the ERP..



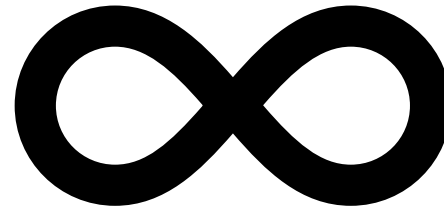
**OFBiz  
based  
solutions**



**We do a project WITH the customer and not FOR the customer.**

## Dev

- Work two by two on a desk.
- Use project management tool (Redmine) integrated with Git
- Power of **plain text!**
  - PlantUML (doc code & business processes)
  - LaTeX (generic docs)
- Every task can't last more than a day / man (**smoother effort less risk**).
- Work on cloud servers (also the mainframe!)
- "**Remote first**" ... in local environment.



## Ops

- Weekly recap. The customer sees what we are doing, without secrets.
- **Customer's awareness.** The client is responsible for what he asks. If it is necessary a rework, delivery time and costs can change.
- The customer has the test environment in the cloud, the client have to dedicate resources to the tests.

You must have a sustainable, lean, effective and completely transparent business model. Use your energy for your core business!

**Table 1: Definition of SMEs**

Company Category	Employees	Turnover	Balance sheet total
Micro	< 10	< €2 million	< €2 million
Small	< 50	< €10 million	< €10 million
Medium - sized	<250	< €50 million	< €43 million

*Source: Commission Recommendation of 6 May 2003 concerning the definition of micro, small, and medium-sized enterprises. (2003/361/EC), Official Journal of the European Union, L 124/36, 20 May 2003*

**Table 2: SMEs and large enterprises: number of enterprises, employment, and value added in 2016 in the EU-28 non-financial business sector**

	Micro	Small	Medium	SME	Large	Total
Number of enterprises						
In thousands	22,232	1,392	225	23,849	45	23,894
In % of total enterprise population	93.0 %	5.8 %	0.9 %	99.8 %	0.2 %	100.0 %
Number of persons employed						
In thousands	41,669	27,982	23,398	93,049	46,665	139,7141
In % of total employment	29.8 %	20.0 %	16.7 %	66.6 %	33.4 %	100.0 %
Value added						
In EUR Trillion	1,482	1,260	1,288	4,030	3,065	7,095
In % of total value added	20.9 %	17.8 %	18.2 %	56.8 %	43.2 %	100.0 %

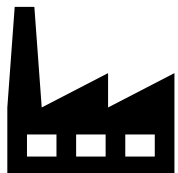
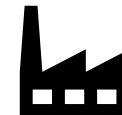
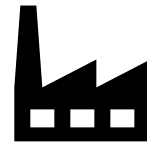
Source: Eurostat, National Statistical Offices, and DIW Econ

Note: Date as of 30 June 2017. Totals may differ from sum of components due to rounding.



- **Why SMEs?**

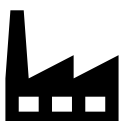
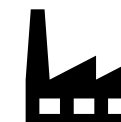
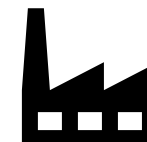
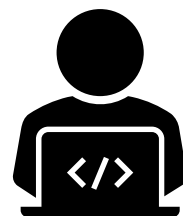
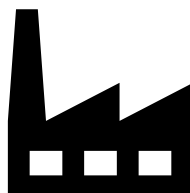
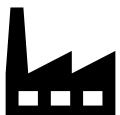
- they are a substantial part of the market.
- The market of big companies requires big software houses and is already firmly in the hands of SAP.
- Of the total ERP installations, in the SMEs the three big players (SAP, Oracle, Microsoft) have a small market share.
- This is an area I have direct experience with. The vast majority of fashion companies are SME.
- They have the requirements of big companies but they do not have the budgets.
- The "software selection" are concrete and point to the requirements.
- SMEs need a vertical solution. They need partners close to them with vertical skills.
- **SMEs are small and "agile"...they need small and agile software and partners!**
- **Open source and the sharing of skills between software houses is the answer!**



Today thousands of (small) software houses work, each with their (big) proprietary Erp.

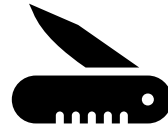


These realities are the IT heart of SMEs. Why this fragmentation? How were they born?



## ONE programming language

- Simple, few commands
- Close to a human language
- Manage video, db, create report



## He **spoke with the users**

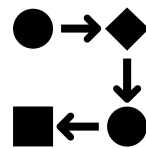
- collect the requirements
- formalized them and wrote the code!



## Strong direct knowledge of the system

- System was developed or maintained internally.

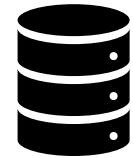
## IT manager



## He knew the business

- Not tech-only, he knew manufacturing & accounting processes

## Mainframe



- Os & Db. All included
- Backwards compatible
- Reliable

## Technological aspect was just a detail

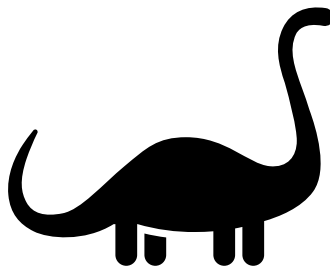
- **IT manager**

- They knew the business processes.
- They had a strong direct knowledge of the system. Often the system was developed or maintained internally.
- They spoke with the users to collect the requirements, formalized them and wrote the code!
- Why? Because **technological aspect was just a detail.**
  - **ONE programming language:** Cobol, Rpg (or something similar) that was incredibly close to a human language. The commands were few and simple. With a single language he could manage video, write and read data from the database, create a report. Everything was a record, Everything was a record in a file. The file could be "display file" or "DB file" or "printer file"
  - **Mainframe.** The operating system was not a problem, it was there and it worked. The database was not a problem, the database was a file. Always, for decades, mainframes were changed but the code was always backwards compatible. The technology evolved, but it was not a problem for those who write code. The programmers simply compiled the code...and it worked 10 times faster! WOW.
  - The requirements and the solution to a problem were important ... not the user interface.

- **ERP solutions were born**

- Every mainframe installation was a "vertical solution", it was all custom, there was no standard product.
- **Former Cto and former mainframe consultant created hundreds of software houses** each with its own vertical solution. ERP born from direct experience in the company. SAP was created by just 5 former IBM employees who started offering custom ERP solutions on Mainframe. Other big names born in sectors other than ERP, enticed by the ERP market, have acquired software houses that had developed ERP solutions. Let's think of Microsoft, which acquired the Danish Navision with 2002, which had been developing a management solution for PC since 1984.
- Natural selection has done the rest over the years, often many "legacy" solutions have been rewritten with more recent technologies, some were born from scratch, starting from specific skills in the sector. New technologies, first client-server, then web. Hundreds of software houses, each has followed its own path, **everyone has written your code, everyone has created his ERP. Everyone has reinvented the wheel, 1, 10, 100 times..**

When I talk about mainframes & COBOL ...  
don't laugh because your bank accounts, your  
insurance positions and much more ... are in  
our hands. Bring respect.



# ERP & SMEs software house: which way?

Technological innovation, competitors, internationalization forces us to choose. *Need to modernize your mainframe ERP?* Many small to medium software houses found themselves at a crossroads....

I have developed my ERP solution over the years and...now?

**I keep my  
ERP solution.**  
My code,  
my customers.

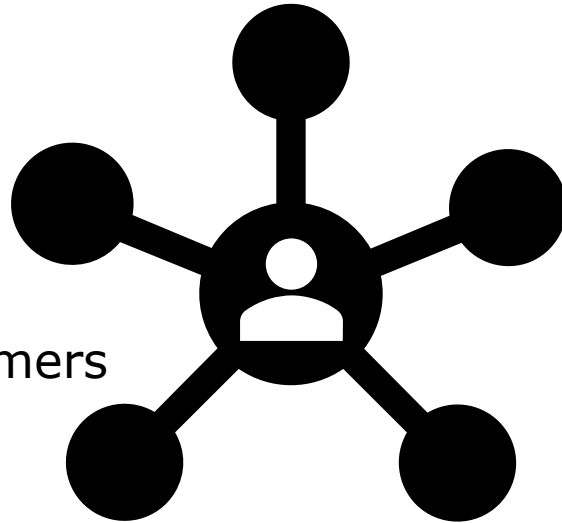


**I become a vendor  
of a great player  
(SAP, Oracle,  
Microsoft).**

*every day we have to choose, not choosing is already a choice*

## ✓ PROS

- **I am the master** of the solution I propose.
- I know the system perfectly.
- The system does everything that customers have asked me
- **I am free.** I decide the cost, the license, the fee.
- I can modify the code without fearing a new release.



## ✗ CONS

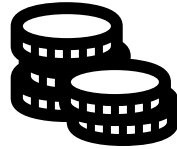
- **I'm alone.** There are no other "partners".
  - I cannot grow in markets, geographical areas.
  - Risky for my customer.
- **Hard tech refactoring.** If I want to rewrite everything on modern technologies a huge effort is needed for a small structure (remember that an ERP is made up of millions of lines of code, hundreds of screens, hundreds of db tables).
  - Often each installation has a different custom code, so rewriting is even more difficult and has very heavy economic repercussions even on customers.





## Artificial Intelligence

- Automates routine processes
- More accurate forecasting (predictive Mrp...)
- Smart UI
- BOT suggestions
- Voice recognition
- ...



## Blockchain

- Smart Contract
- Distributed ledger
- ...



## Device independence

- Mobile
- Notification on smartwatch
- ...



## noSQL

- Fine grained logging for security and data auditing
- Review of some data intensive task
- ...



## Cloud

- Serverless / Microservices
- Faas (Function As A Service) for seamless integration
- ...



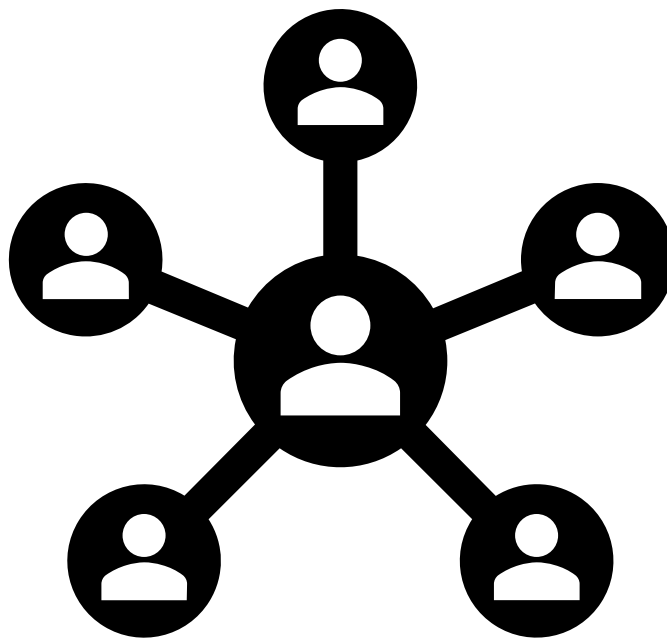
## Internet Of Things

- Monitor machine operations
- Social analytics and a more direct interface with customers
- Sensors and devices: intelligent recommendations for supply chain managers

...do you REALLY think you can stay up to date?

## ✓ PROS

- I am part of a "large family" of international partners.
- I have the **support of the parent company.**
- I can take advantage of an **international and well-known brand.**
- The software core is international, standard, written in a modular way.
- The parent company thinks of technological innovations, **I care about business** not about tech.

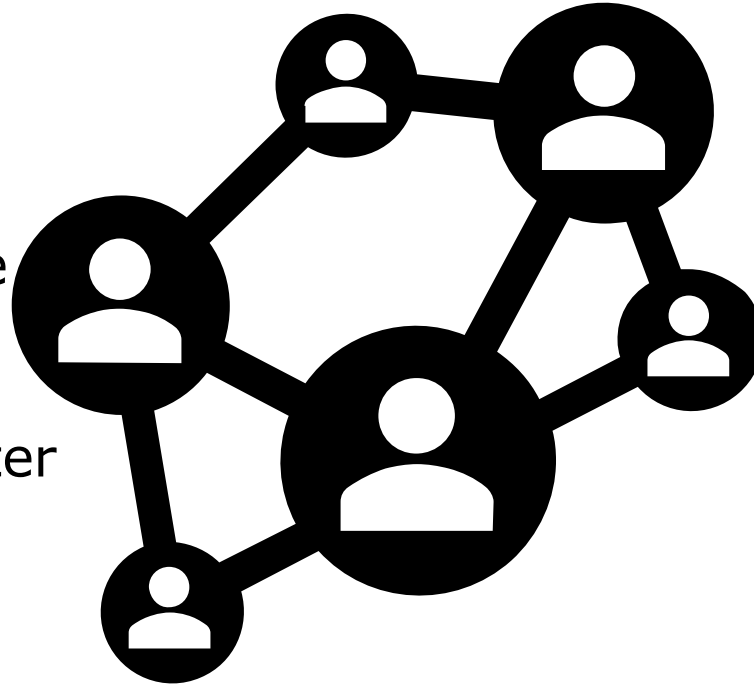


## ✗ CONS

- The software was not born for a sector, **certain business logics MUST be made from scratch**, the data model is not necessarily supported.
- Often **partners enter into competition with each other.**
- **I am not the master of the solution I propose.** Prices "imposed from above". I find myself developing my own solution but it risks being "out of the market".
- **The technological directives are imposed**, I have to adapt. I have to follow the courses, if the technology changes my code may have to be rewritten.
- Another business partner could join a "my" customer. The customer is the big player's, not mine.
- The "core" software is closed.

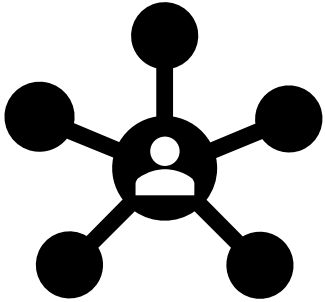
## ✓ PROS

- **Exiting!** The experience of working with people from other cultures and cutting-edge technologies. Working in "remote team".
- Open solutions follow **technological innovations** faster than a small closed solution.
- **Open code**, maximum freedom.
- I'm free. **I can decide prices** independently.
- **Sharing**. Other developers and software houses around the world use the same solution.
- The software has an international appeal.



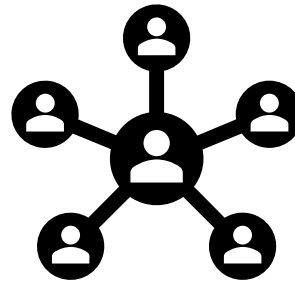
## ✗ CONS

- **I don't have the brand** and the appeal of the big brand.
- Open source may not yet have a good reputation in companies (for ERP solution).
- I still have to "change" the platform and **rewrite my code**, my verticalization on a new platform.
- However, depending on the community, if the project is closed it can be a problem for the survival of my product.



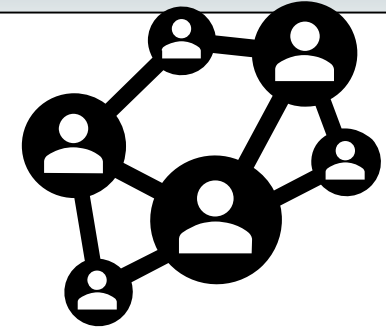
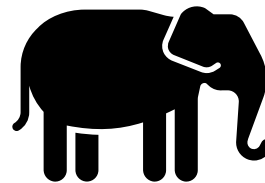
## My proprietary ERP

- I'm free
- I'm alone
- Reinvent the wheel
- Difficult to keep up with technology



## Partner of a big player

- I'm not free
- I'm not alone
- Big brand label
- Big brand tech



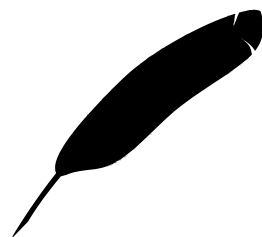
## Open Source ERP

- I'm free
- I'm not alone
- Remote team, cutting edge-tech, sharing!





There is a simple way to capture some types of monkeys. Take a banana and put it in a wooden box. Make a slit in the wooden box big enough to let you pass the open hand of the monkey. The slit must be small enough not to pass the banana. The monkey puts his hand to take the banana, he doesn't want to loose his treasure, so he doesn't leave the banana and stays there. At this point it is easy to capture it. He loses his freedom for a banana. Could it be that **many small software houses like little monkeys lose their freedom so as not to lose their little banana ERP?**



Apache

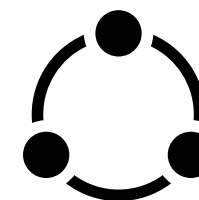


Java

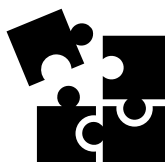
## Why OFBiz?



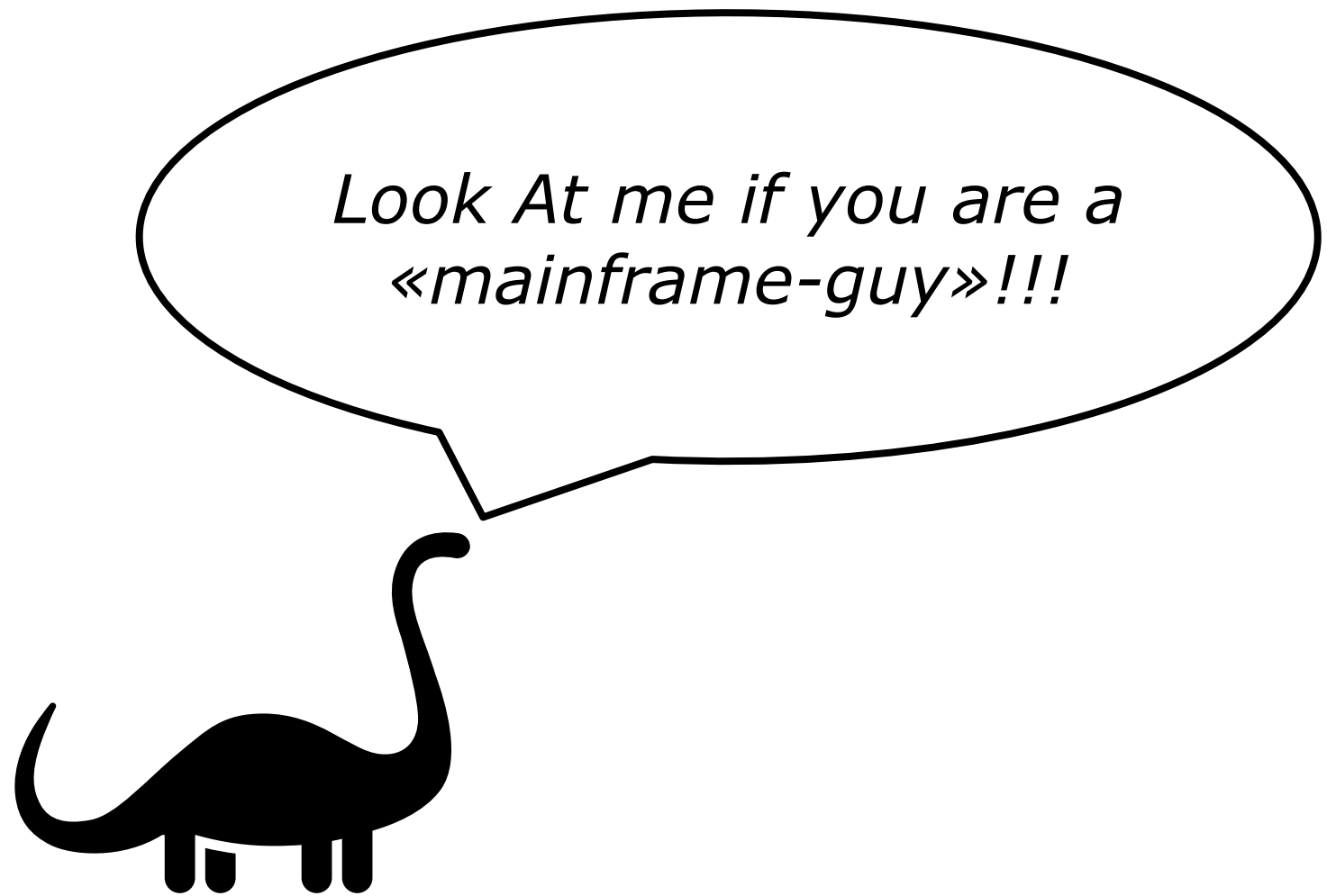
DB independent



Framework



Modules





The Apache way!

- **Apache is Apache.** What should we add? It is a guarantee, it is a brand!
  - Therefore it automatically gives credibility to your custom ERP.
- The **community** is solid, organized, seriously structured.
- Then the **Apache license** is much better suited to an ERP business project. The product can be branded and licensing can be managed more freely. We can use, customize, extend, modify, repackage and resell completely free of charge.



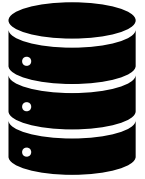


**Java is THE enterprise language** par excellence

- Java has an excellent reputation.
- Java is a **widely used** language, it is possible to find a programmer locally or remotely.
- Java **can be dynamic** using Groovy
- Java has the support of Oracle, Ibm, SAP.
- Java has an almost infinite **library ecosystem**



*If you work on mainframes you probably know java*

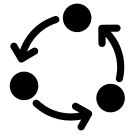


## Db independent.

- Probably your ERP solution is based on a specific database
  - **You can use the database you've always used**...you can use an open db or a proprietary db, without problems.
- The **logic remains ONLY at the application level** and is not fragmented between the database and the application layer.
  - Stored procedures or triggers are not used, written for example in TSQL or PL/Sql



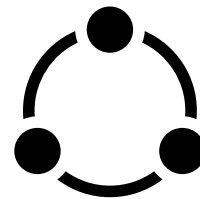
*Simple implementation also on mainframe's DB2*



MVC



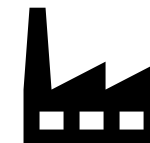
OS rather than a  
framework



**OFBiz framework is  
"developer oriented"**



XML based



Born for ERP



## Amazing framework. OFBiz is “developer oriented”!

- OFBiz is an operating system rather than a framework
  - Like when you were working on mainframe, you can really **concentrate on the business processes**, you don't have to worry too much about what's going on below.
  - Is NOT the usual “general purpose” java web framework. It was **born specifically for ERP**, Crm, e-commerce. Without too many frills, a lot of substance.
  - **XML based**. In a “pragmatic programmer” approach...you can always parse an xml and convert it!
  - MVC

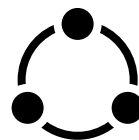


*Batch processing, scheduler, job queue, wrkdbf...like mainframe's os!*



## Controller

- Xml based
- Java or Groovy

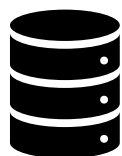


## OFBiz M.V.C.



## View

- Xml Based
- Modular widget system
- Freemarker Templating



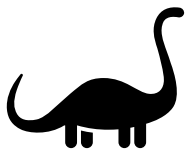
## Model

- Xml data def
- No ORM!



## MVC - Model

- Is the definition of the tables sql? No, **always and only XML**. The entity engine thinks about generating the tables and modifying them by adding fields if the definition xml has been modified.
- **No ORM** that generates unwatchable runtime queries, which kill the db's performance. Let's talk about 500 database tables. The tables have records. **A record is a data structure, it is not an object!** A record is an object without methods, it has only properties. But then? NOT an object. Wanting to forcefully think of an object is a stretch, we work on a database, we work on records. Thinking of them as objects creates a useless mental and technological superstructure. If they were objects we would use object-oriented databases, but no, we use relational databases.
- *Jira is used for issue tracking and project management by over 75,000 customers in 122 countries....Jira use Apache OFBiz entity engine*  
*Xml entity def is like PF/LF DDS. The GenericValue in OFBiz is like a DS!*



## MVC - View

- **Xml Based.** The basic definition of the screens and widgets is also in xml human readable even for non-tech people. The additional level of detail is Freemarker, which is a fantastic templating language, extremely simple and readable with a very low learning curve. The further detail is html-javascript code.
- **DRY.** Don't repeat yourself. In an ERP there are hundreds of screens, a modular widget system is very important.



*Screen definition in xml is very like the DSPF or the SCREEN SECTION in Cobol.*



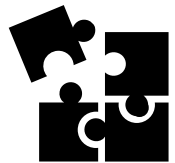
## MVC - Controller

- **Xml Based.** The definition of the controller and services seems a workflow management language. Without having to read java code, it is already **possible to understand the flow of information**, also for non tech people. Very new code is written in Groovy, using a fantastic DSL language that is very easy to read.
- OFBiz is **natively service oriented**. It is not the classic monolithic ERP! OFBiz is born lean, born for the cloud, service oriented.



*you can think of each service as a Rpg/Cobol program with input parameters*





## Modules

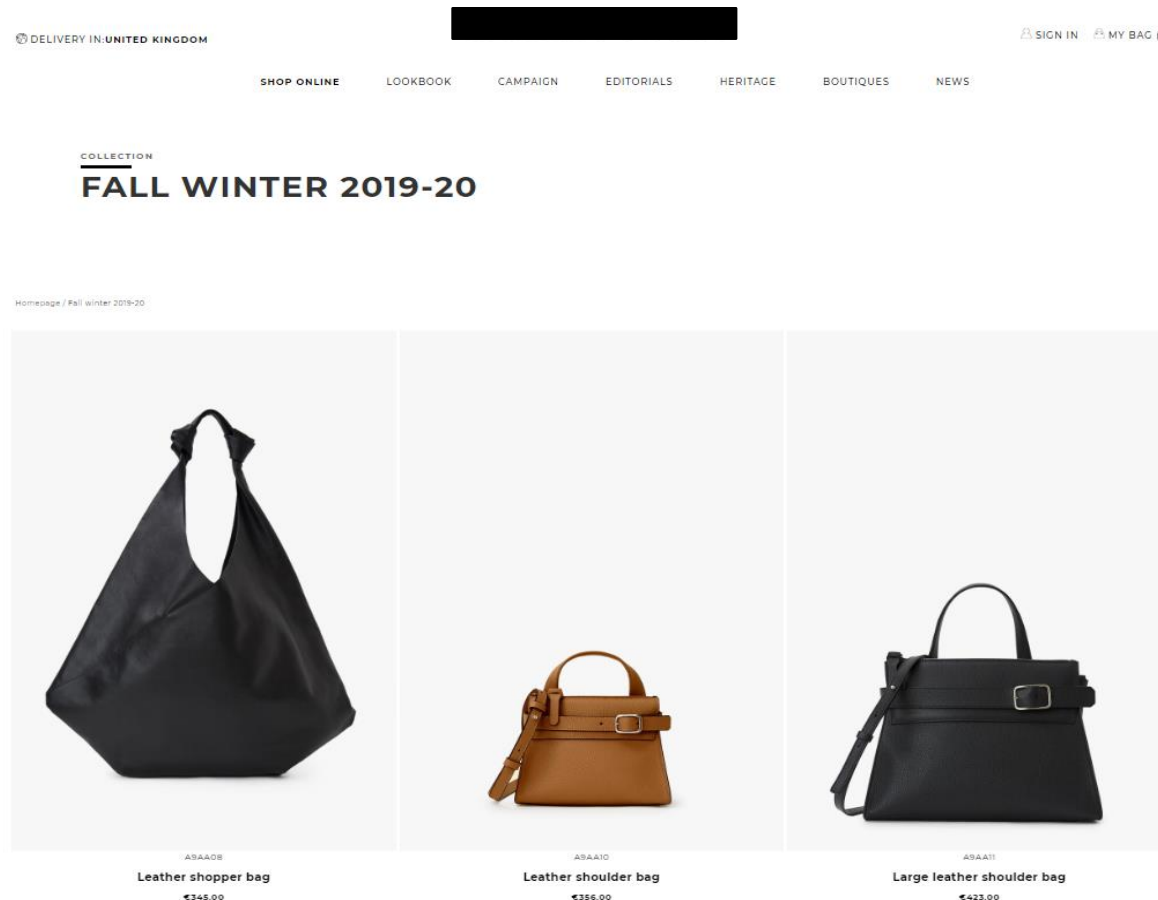
- DRY! A lot of **modules "ootb" or "ready to customize"** (e-commerce, invoicing, customer, product, mrp...). We must not reinvent the wheel!
- **Plugin system** allows you to build your own modules, customize them to create your own e-commerce management, your own product management or invoicing.
- If the "out of the box" management does not meet all the business requirements the infrastructure is good enough to build your own.

...I found myself in need of developing a B2C solution. I could choose one of the many open source e-commerce frameworks available...But...

- **What makes me unique?**
  - Experience in business processes
  - Experience in ERP systems
  - ...That's why I chose OFBiz... not because it was the easiest way to create an e-commerce, indeed, but surely it is **the most effective way to develop your ERP**

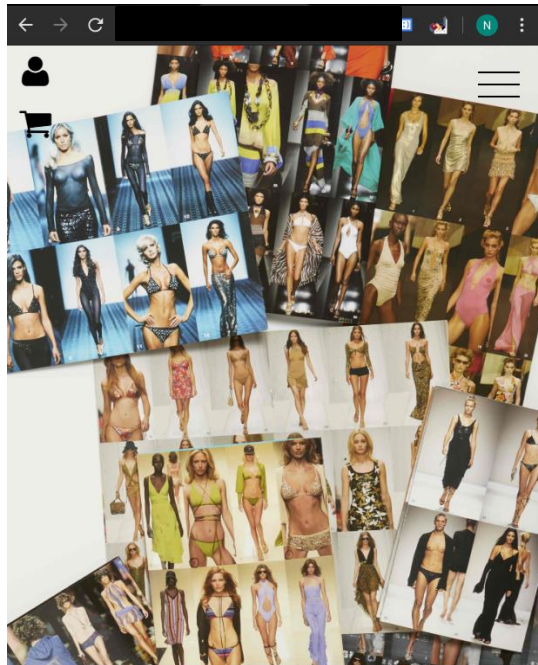
- **Front-end from sketches**

- Mobile-first, Bootstrap, Ajax, Different UI for different brands, but the same backend



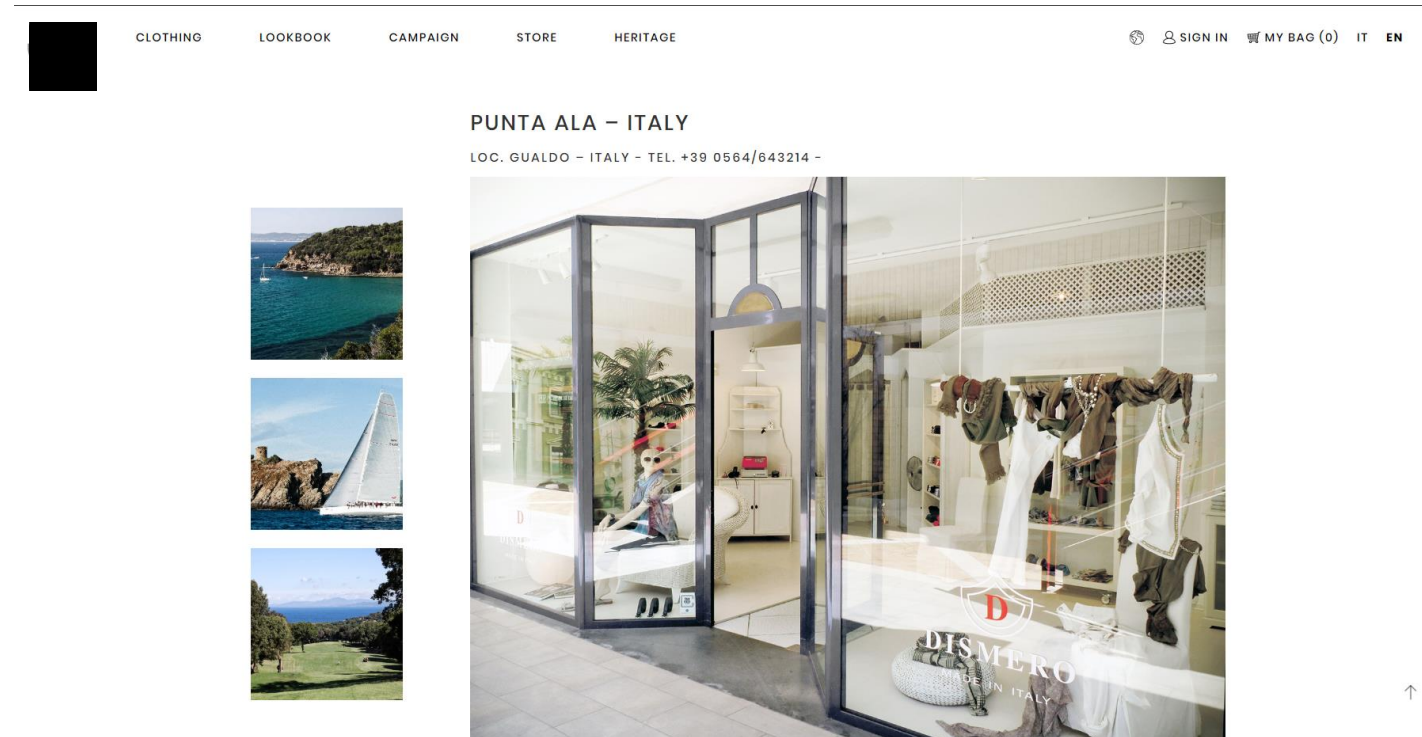
- **Content Management**

- We did not rely on external systems or tools.
- On the side OFBiz we have developed a simple tool to populate the backoffice content, using the OFBiz infrastructure, without touching the standard entities.
- We chose to concentrate everything on OFBiz, from "ERP-guy" we thought to **simplify and standardize everything on a single platform.**



#FASHIONSHOW

Dal 1997 [redacted] partecipa a sfilate di Milano Collezioni presentando i suoi modelli con le più importanti modelle internazionali e testimonial di rilievo.



- **Color-Size Matrix in a nutshell**


- OFBiz natively manages "real products", "virtual products", "features". Without touching the data model we managed the Color-size matrix. When we think of buying a hat we can't just buy a hat, we have to decline it in the real world; the "real" object must necessarily have a color and a size. In short, the "model" of hat is a virtual product which, to become real, must be associated with a color "feature" and a "feature" size. The colors and sizes create a matrix. Each box in the matrix is a real product. The real product is a variant of the virtual product.

<i>Product: HAT</i> <i>isVirtual=Y</i> <i>isVariant=N</i>	<b>Feature: S</b>	<b>M</b>	<b>L</b>
RED	<i>Real Prd: HAT-RED-S</i> <i>isVirtual=N</i> <i>isVariant=Y</i>	HAT-RED-M	HAT-RED-L
WHT	HAT-WHT-S	HAT-WHT-S	HAT-WHT-S
BLK	HAT-BLK-S	HAT-BLK-S	HAT-BLK-S

- **Color-Size Matrix..the final result**

- **B2C.** The final consumer buys 1 piece in 98% of cases. Cool UI is important!
- **B2B.** Who inserts a "seasonal order" inserts hundreds of lines, buy dozens of pieces and must have a dedicated ui

Home / XXXXXXXXXX



**SKU**  
103054  
**Nome prodotto**  
ABITO LINO  
**Descrizione breve**  
ABITO LINO

**Descrizione lunga**  
20120  
**Composizione**  
100% lino

	40	42	44	46	50	52
<b>BEIGE</b>	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €	<span style="color: red; font-weight: bold;">out of stock</span> 118,00 €
<b>BLACK UNI</b>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

[Aggiungi a ordine](#)

SECURE PAYMENT - ALWAYS FREE RETURNS

☰
**LIVIANA CONTI**
🔒 (0)

Questo jumper ha una slim fit. È fatto di stretch viscose e ha un collo rotondo, maniche corte con cuciture orizzontali, rifiniture a coste e arriva alle hips. Fitting: SLIM

**SHIPPING&RETURNS** +

---

**MATERIALS** +

---

**SIZE** 40 ▼

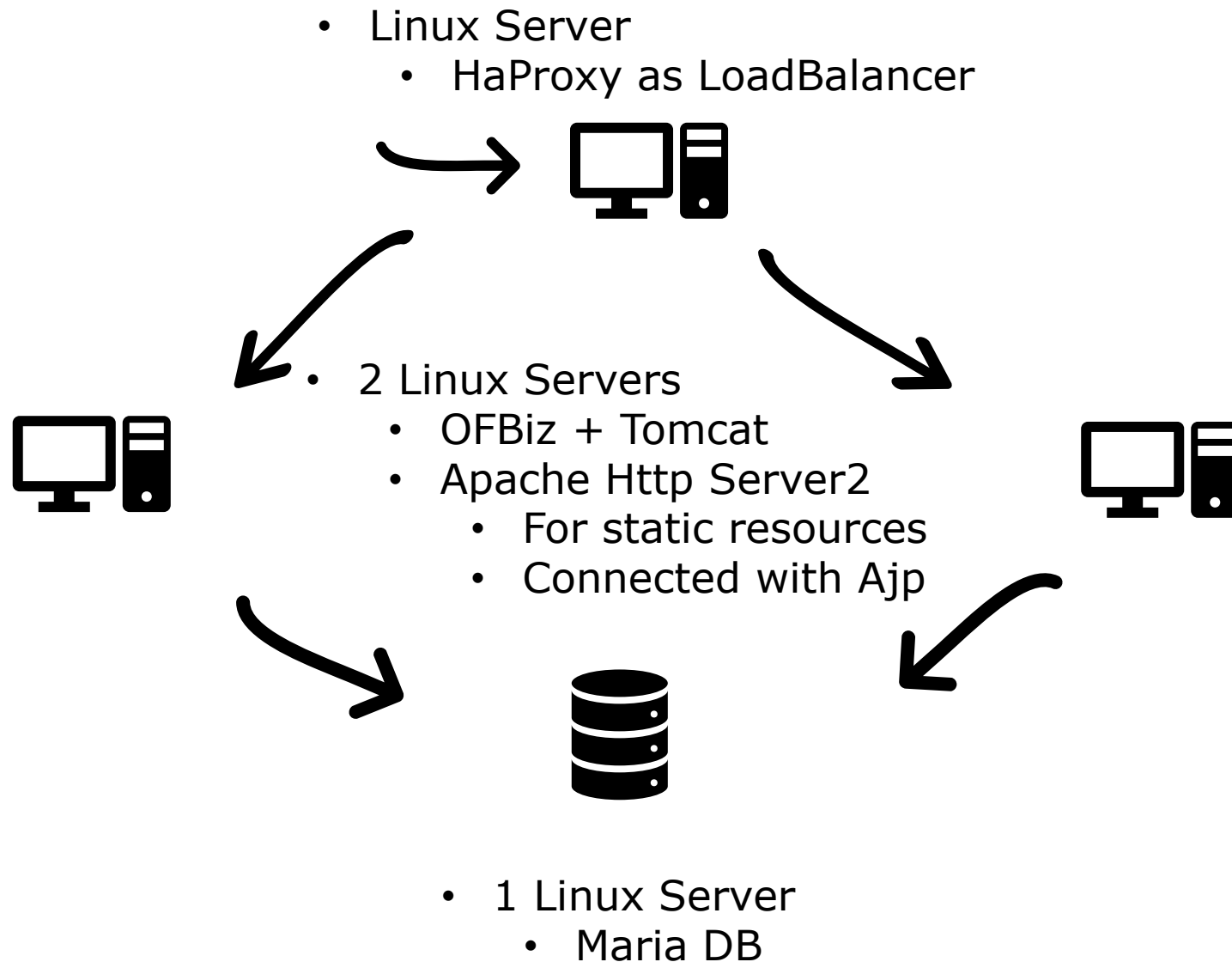
---

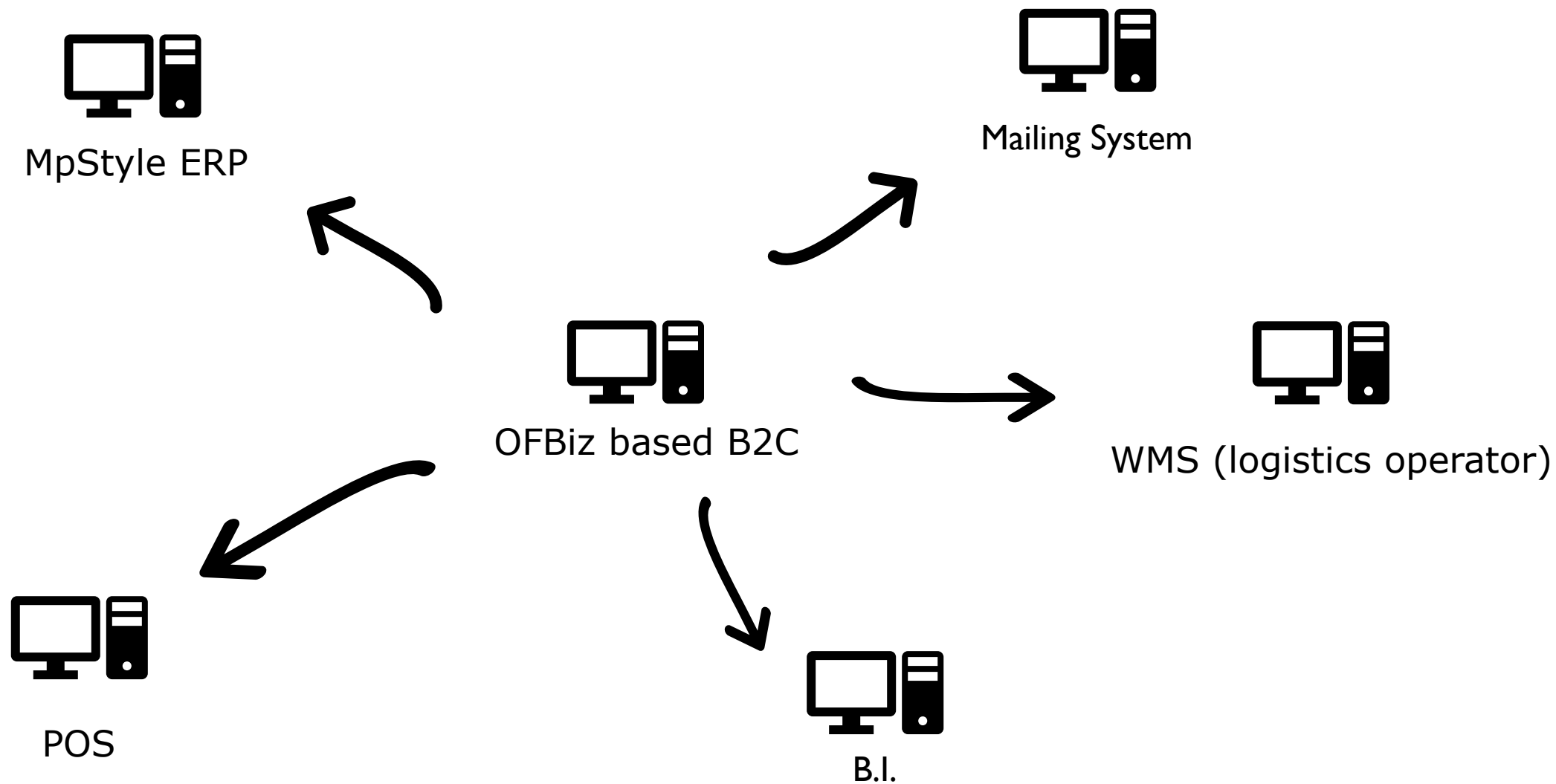
**COLOUR** 
Gesso-Oceano ▼

---

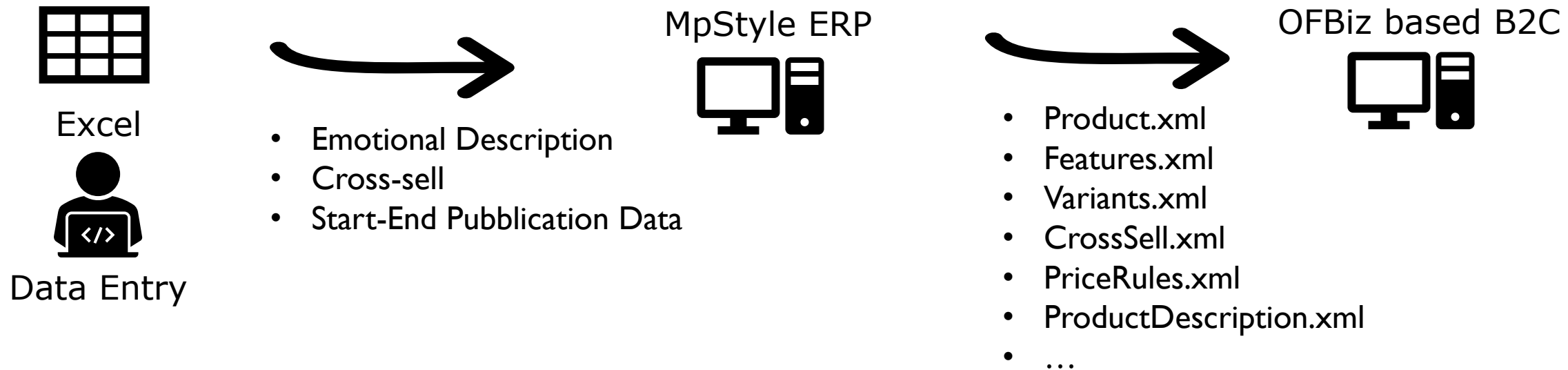
[SIZE GUIDE](#)

ADD TO BAG



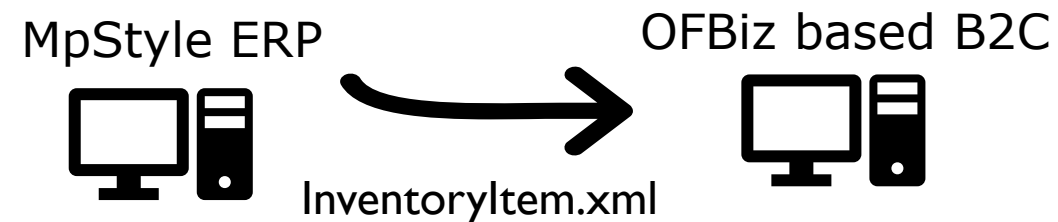
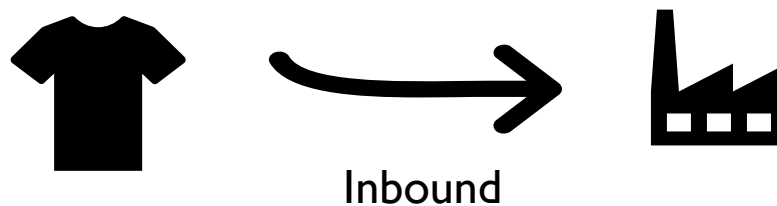
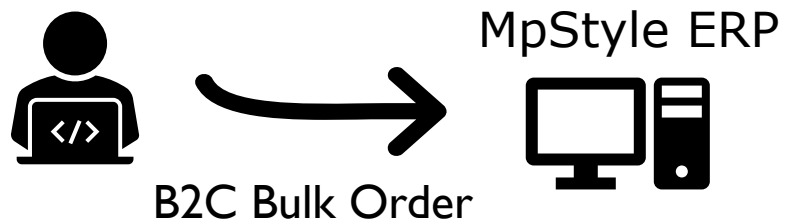






## • Publishing Tool

- We have developed what we called the "publishing tool" on the ERP side. The operator can integrate the data already present in the ERP (codes, descriptions, price lists ..) with the missing data **via import from excel or through manual data entry**. We talk about emotional descriptions, descriptions in language, cross-selling, start and end publication data. When the elements are complete, **the user can select the article as "ready to present" and the data is provided in XML** format on an SFTP server. Similarly, **a service programmed on OFBiz imports the data provided every hour.**



## • Stock availability

- The company buyer places a bulk order on ERP at the beginning of the season, in accordance with the budget and sales projections. At the end of the production cycle, the garments enter into logistics and through ERP the e-commerce order is automatically assigned for the relevant garments.

OFBiz based B2C



Invoice.xml

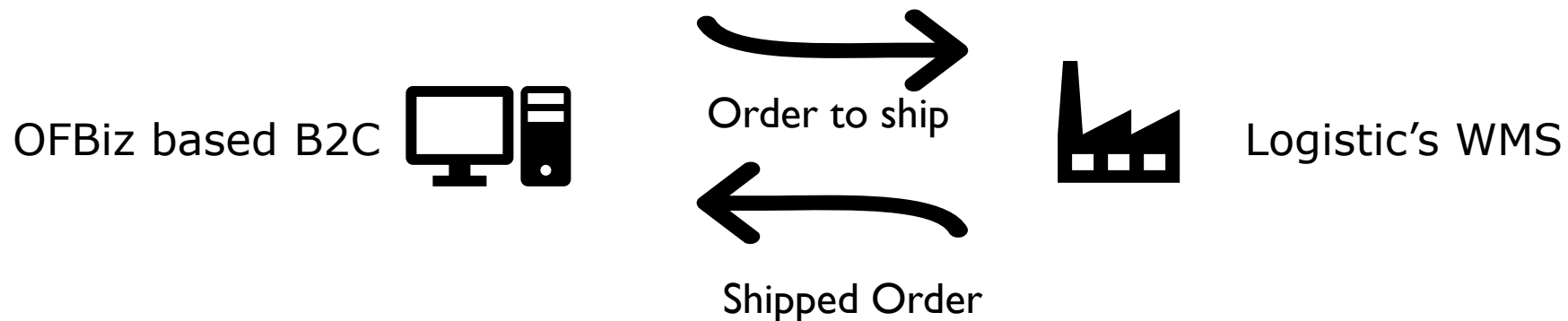
MpStyle ERP



- Accounting
- Update Inventory

- **Shipped Order**

- Each time an article is sent on B2C, **the ERP receives the flow of the shipped orders**, so as to download the warehouse (or load it in the case of returns) and make the accounting entries



- **Logistic**

- The customer has delegated warehouse and shipments to a third party, in order to concentrate on the core business. A seasonal business requires considerable flexibility that only a specialized operator can provide.
- Each time OFBiz B2C receives an order, it is given to the WMS that handles the shipment. The Wms sends OFBiz all the shipments made, a special OFBiz service acquires the shipments by completing the process, issuing the invoice, sending the email to the customer for shipment...

## What's omnichannel in retail? (focus on inventory...)

*“Omnichannel retailing is an expansion of multichannel retailing. The emergence of digital technologies, social media and mobile devices has led to the significant changes in the retail environment and provide opportunities for the retailers to redesign their marketing and product strategies. Today, customers tend to be looking for information in the physical store and at the same time they are getting additional information from their mobile devices about offers and possibly better prices. Omnichannel allows organizations to allocate inventory availability and visibility across locations vs. each channel holding specific units. A number of features, like size charts, easy return policy and same-day delivery, have boosted ecommerce and promoted omnichannel shopping” (from <https://en.wikipedia.org/wiki/Omnichannel>)*

## **Omnichannel Order Fulfillment**

*“Omnichannel Order Fulfillment is a material handling fulfillment strategy and process that treats inventory as fully available to all channels (e-commerce, store replenishment and wholesale) from one location. While the internal fulfillment process may diverge to optimize the operations, the outbound process only diverges at the point of pack out and shipping.*

*It allows companies to deliver goods to multiple channels using a single facility with a single workforce, delivering from a single inventory. The channel dictates the order size, delivery requirements, packaging, shipment method and rules for handling shortages. Single-channel facilities can have a fixed set of criteria...”*

## **Omnichannel Flow Paths**

- *Order online/deliver to customer*
- *Order online/pick up at store*
- *Order online/deliver from store*
- *Order online/return to store*
- *Select and pickup at store*

*([https://en.wikipedia.org/wiki/Omnichannel\\_order\\_fulfillment](https://en.wikipedia.org/wiki/Omnichannel_order_fulfillment))*

## What’s “click from store” ?

The customer orders in store but using e-commerce, through "assisted" sale.

## Why “click from store” ?

Why, in fashion industry, is this solution among the different possible in omnichannel optics so strategic?

- Because allows you to keep few products allocated in the individual store;
- Because for **luxury brands the locations have very high maintenance costs and the exhibition space is reduced accordingly**. In fact in the concrete case the customer had the problem opening a shop in Venice, near Piazza San Marco, where the cost per square meter for rents is important;
- Because the trend requires that the display of goods be minimal, without over-exposure of products. **Shelves full of clothes have a cheap feel**. All the big fashion houses try to thin out and expose only the significant items;
- Because it is **impossible to have all the products for all sizes and colors**;

## How was the "click from store" implemented technically?

- **A dedicated webstore (with real-time sharing of central stock availability)** accessible only by the sellers was developed on OFBiz. The special section shows the availability of the central warehouse; therefore every time the user click on the product page, OFBiz calls a ws-rest on ERP (input: Sku out: QuantityOnHand) and aligns the inventory item. The same operation is done at checkout.
- **In the checkout phase only "Pay Offline" can be selected; the order is transferred to the POS software via ws.** Therefore the order on the POS issues the receipt and collects the sale. In this phase a ws-rest on POS warns OFBiz that the order has been paid.
- As in a normal online sale, when the payment gateway completes the order this is sent to the WMS software to complete the normal shipping process.

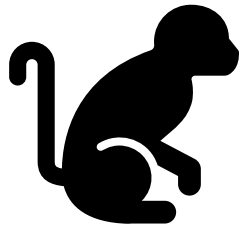


- “Soft migration” from your erp to OFBiz
  - use a lean and sustainable approach
    - Be DevOps and keep calm. Rome wasn't built in a day. There are thousands of lines of code in your ERP.
    - Spring Cleaning and refactor! Review your data and processes. Migration is not only “copy-paste of processes and data”!
      - Involve customers who currently use your ERP. You don't always know the requirements better than them.
    - Learn about the standard services and modules of OFBiz. Don't reinvent the wheel!
    - Small tasks! Migrate one module/service at a time. For example start from the "bill of materials"
    - Progressively integrate Old & New ERP. Use microservices, message broker (Apache Camel, RabbitMQ)
    - If possible install OFBiz db on your ERP's dbms.
    - DRY! Think about the services and screens typical of your business. In my example, the size/color matrix management
    - If possible write a parser to convert your old screen to OFBiz Xml screen def!

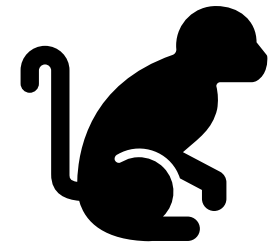
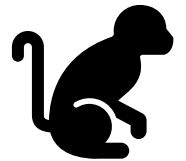
"I found a fantastic job! I'm writing some crazy code, I'm working with Kubertenes and Docker, in Python I'm developing a microservices structure, in my team we use Git, the deployment we will do in the cloud on Aws, all serverless... unfortunately I only have a contract for 6 months and the salary it's not that great ..."

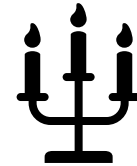
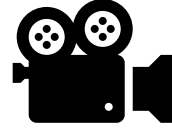
"Ah, I have a good contract and the salary is very good ... but we do ERP software"

**Open source can make ERP exciting!**

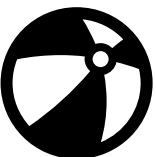
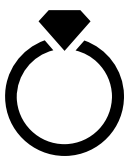


The various “business partners” of OFBiz can create an ecosystem of verticals based on specific realities, helping to improve the data model, the framework and the business modules. This is a sustainable way of working, developing with the community. I think that open source can be a great resource for the ERP world, for small-medium companies, for emerging countries, but also for the various departmental software of large enterprises.






**Every object that surrounds us**, every object created in a company, after being designed and drawn has become a BOM. **It was "created" within an ERP** system, was dismantled into its components and generated a Mrp, its components traveled and passed through other companies, other ERPs, other people, sales orders ... purchase orders ... receiving goods ... supply chain, order fulfillment. **All the processes that govern goods and services are modeled in ERP**, are born in our mind and become business rules written in an ERP. These rules may not be modern microservices, perhaps they are not in the cloud; but those are the rules that govern goods and services all over the world.



# Thanks!

**Questions  
and (hopefully)  
answers**



Bye bye,  
I go back to  
your bank