

# I IOT SURVIVAL COURSE IN THE MANUFACTURING VALLEY

RICCARDO ZAMANA



# INTRO

# MY INITIAL SCENARIO

- New Acronyms (HMI, SCADA, MES)
- Proprietary Protocols
- New Vendors
- Cloud is a prohibited word
- Internet is a prohibited word
- Many Overlapping Architectures



# FIRST 6M EXPERIENCE

- Many «Language» tentatives
- Many Architectures Big Picture
- Many problems in the field
- Many meetings with «Leaders», «Gurus», «Evangelists»



- Who's in front of me? **A person who exhibits a problem that it IS NOT an IT problem**
- Before judging, or thinking HOW to solve the problem, **we need to understand WHO is in front of us**



# THE PROBLEM

## FIRST: UNDERSTAND THE MARIO'S STORY



# “WORKING HARD, TO KEEP AT THE WEEKLY SCHEDULE”

Why do I need this data?

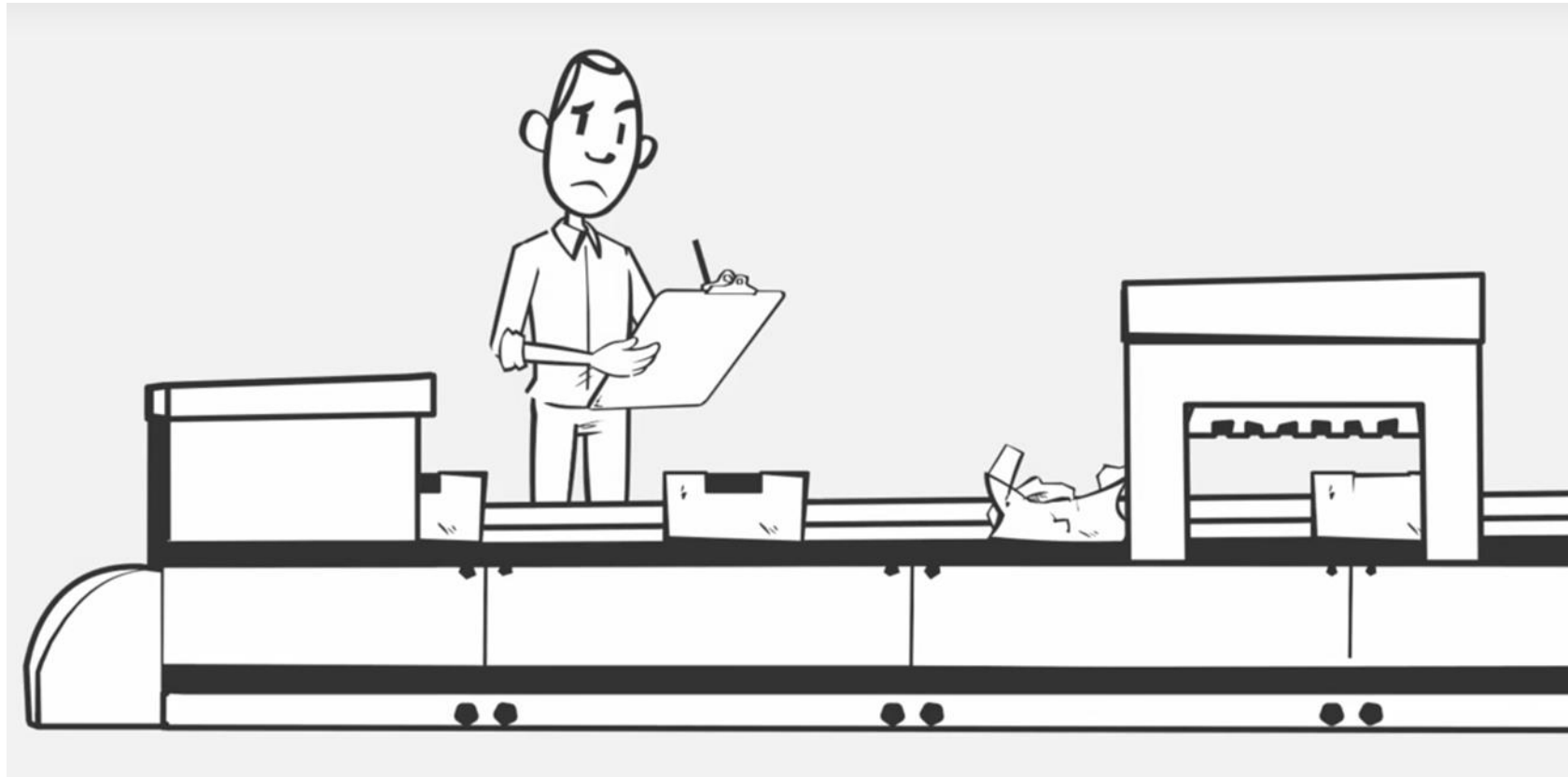
Because I have to **GUARANTEE the production.**

Production must not be delayed and must not have waste ... because otherwise I am consuming raw materials (paid) more than necessary, so the gain decreases (more raw materials, more hours of work).



# WHEN A PROBLEM OCCURS

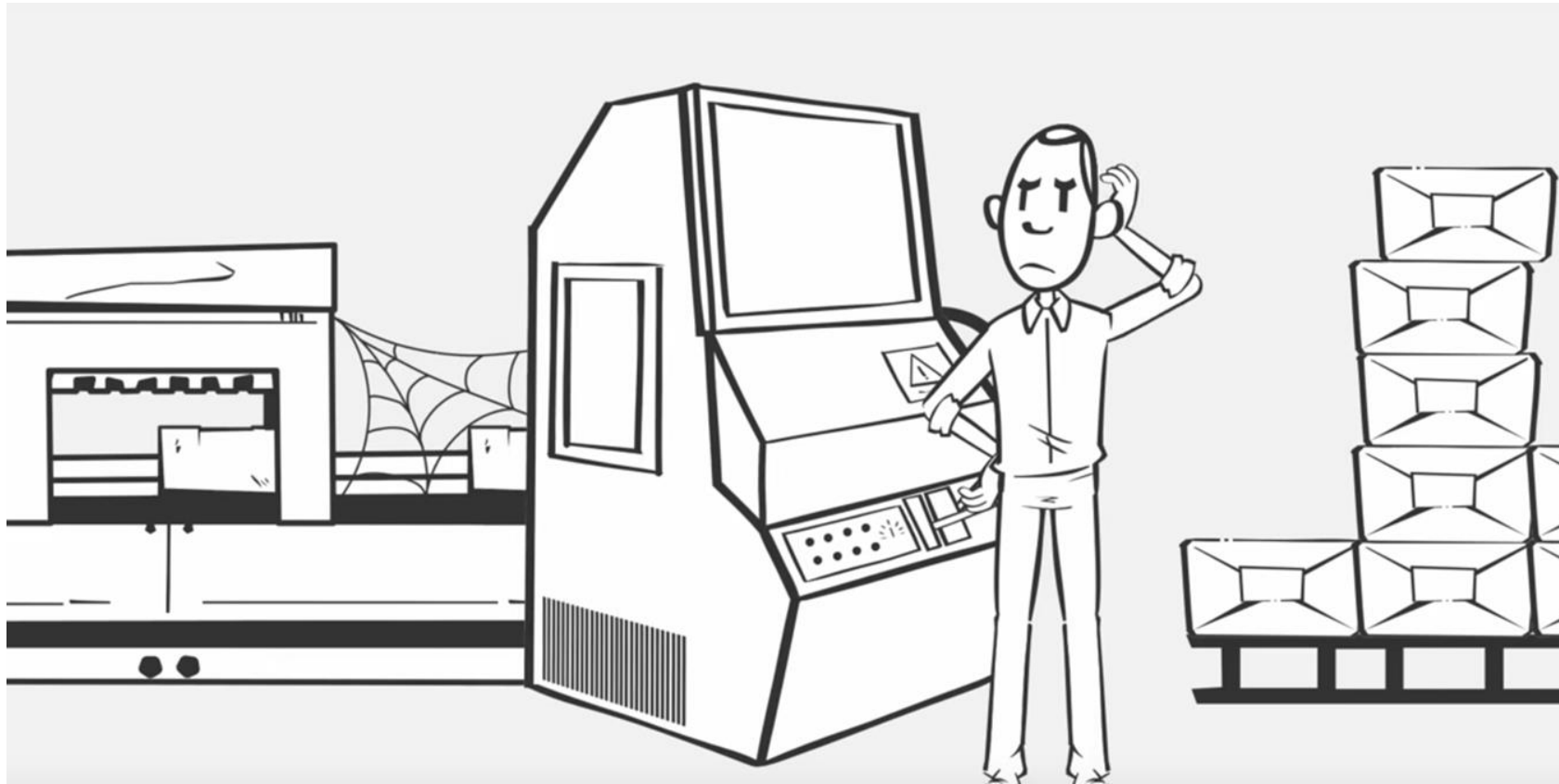
- It is not possible to understand the initial cause of the problem in a scientific or heuristic way.
- "Theoretical" production is hardly close to "real" production. Because contingencies are neither predictable nor governable





# ASSETS ARE NOT EQUAL, AND NOT NEW

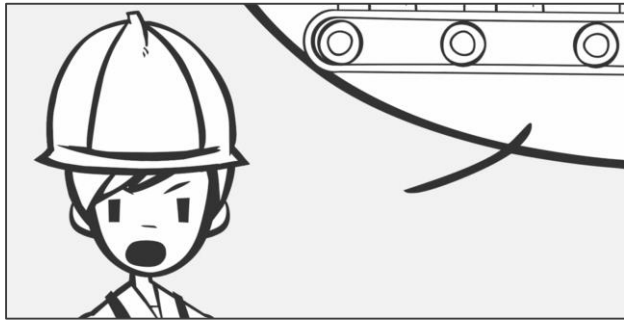
- Often PLC Data is only about Pieces, but nothing else
- Machine Producers propose innovation.. but what of it?



MARIO IS BEATEN BY EVERYONE. EVERYONE IMPUTES A DIFFERENT CAUSE.

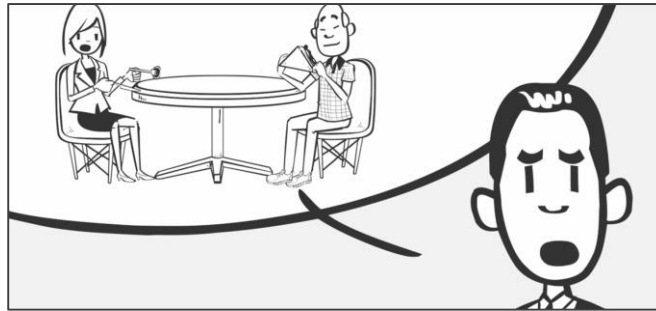


## SOME EXAMPLE

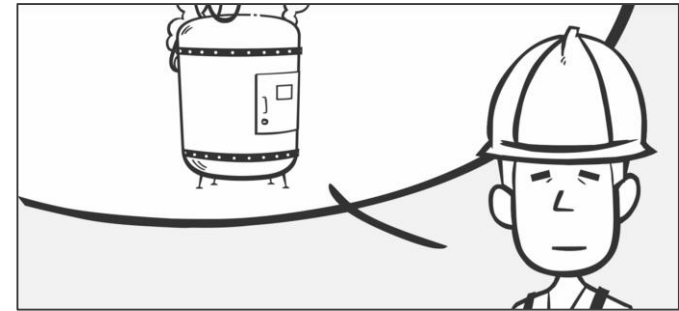


High-speed machinery does not guarantee the same performance ... and we must reach the limit

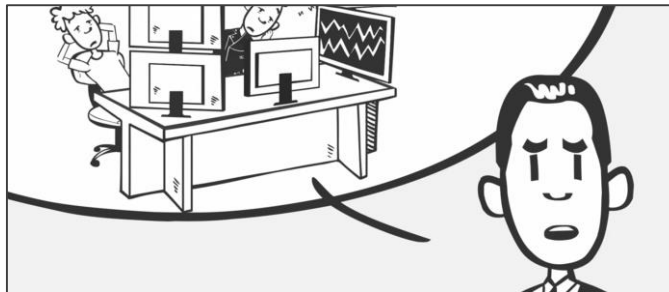
The rollers have very short stops so they have a discontinuous movement



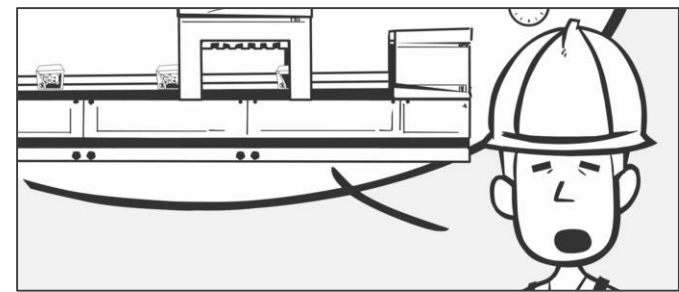
Too many long Lunches?



«Suffering» machinery



Night shifts not efficient



Format Change too slow

# CONCLUSION

- He doesn't know how to start
- He's asking to solve an operative problem (not theory or big pictures)
- He can't give you NRT data, or historical data
- From ITALIAN: **Mario è «incazzato come una biscia»** già dopo 5 minuti che non vi capite.



... Ed è **OVVIO** che a Mario del «CLOUD» non gliene frega niente.

# REAL GOAL

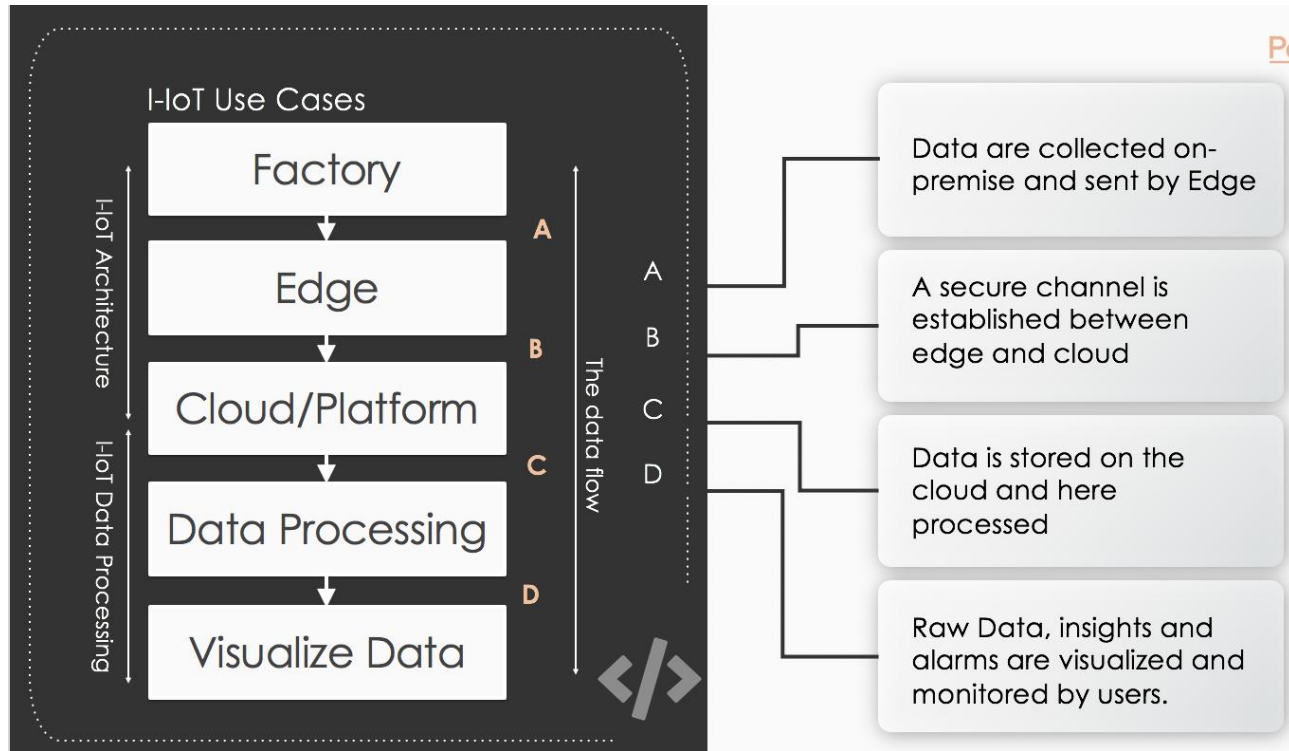
- Mario needs to be able to understand what is happening, looking at the data from his tablet, from his excel, from the phone
  - Must be able to look at the statistics / aggregates and download the RAW data to watch them in detail or compare them only in case of need
  - He needs to DEMONSTRATE solutions/facts to the others stakeholders
- 
- 1) GIVE BACK CONTROL to MARIO
  - 2) REMOVE STRESS from MARIO

BACK TO SCHOOL

## TYPICAL USE CASES

- **Highly customized products**; the goal is to have the single consumer customize the product according to his needs
- **Asset Performance Management**: Industrial equipment monitored and their performance controlled
- **Condition Based Maintenance**: Equipment maintenance not according a fixed schedule but according to their real condition; less cost and downtime with more productivity
- Personalized services and new business model => **SERVITIZATION**

# IIOT 5 PILLARS

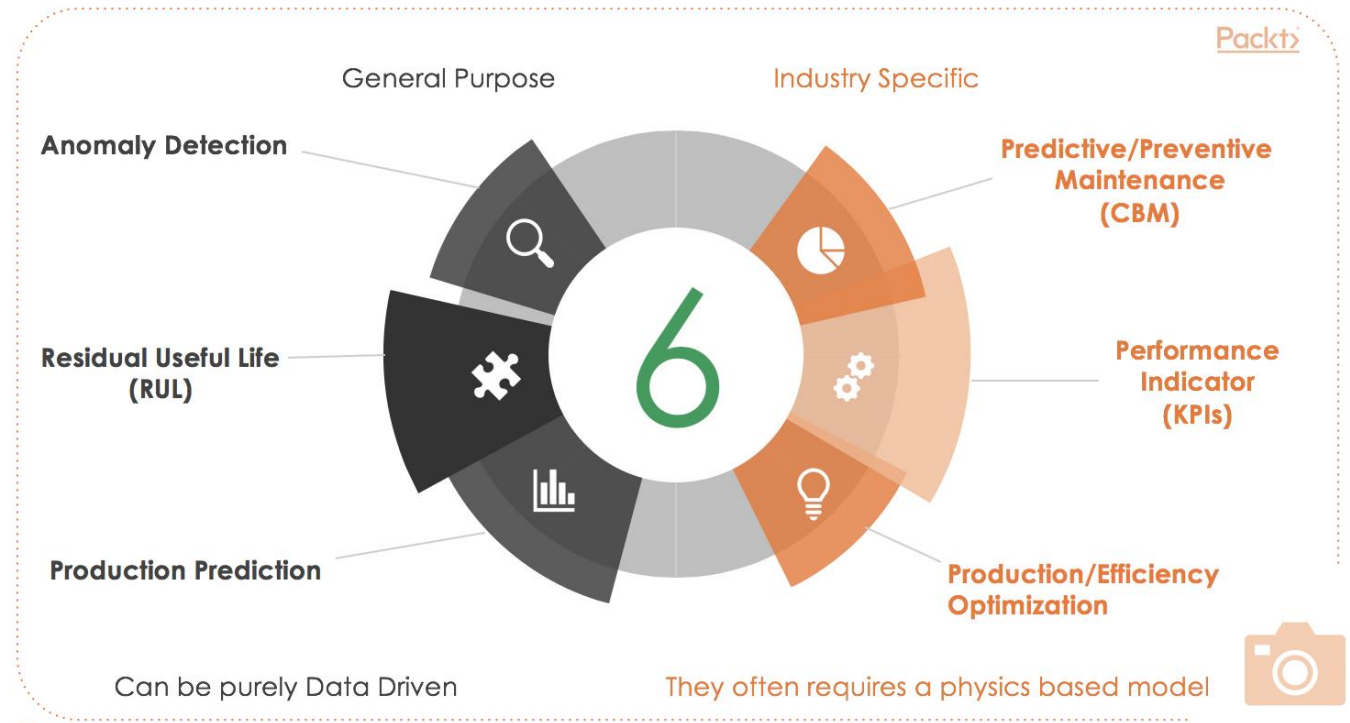


- Collecting
- Sending
- Storing
- Processing (thru analytics)
- Visualising data.



# IIOT ANALYTICS

- **Anomaly Detection**
- **Production Prediction**
- **Residual Useful Life**
- **Optimisation**
- **KPIs**
- **Predictive or Preventive Maintenance**



# THE REAL LIFE

# THE HAWAY PIZZA PROBLEM

- Flexibility is required in industrial projects
- Cloud / Fog are always present
- There is a lot of fear because there is no knowledge

...REMEMBER...

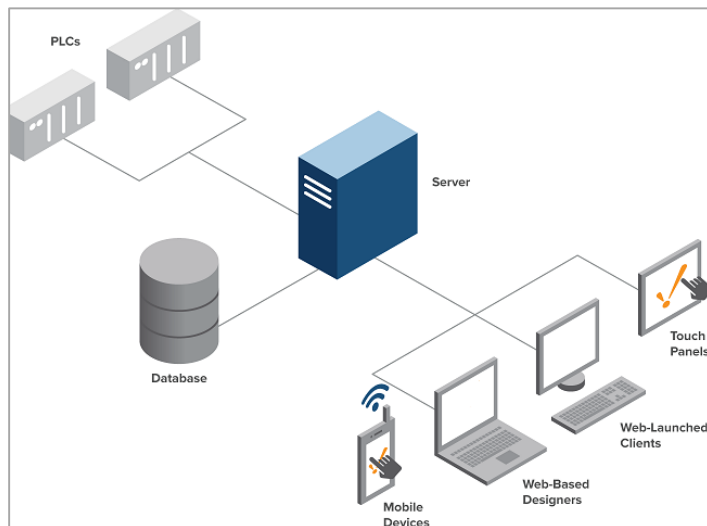
*«Voglio tutto quello che mi hai  
proposto... ma da qui non esce  
niente»*

*Cit. #UnClienteVeramenteAdorabile*

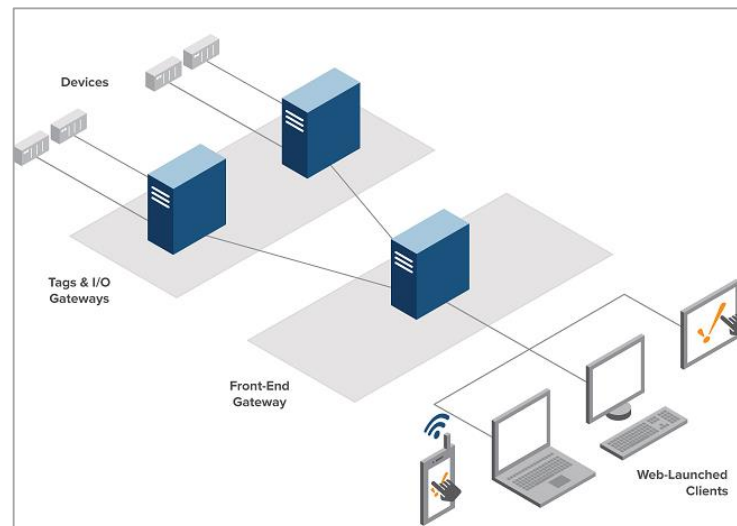


**CLOUD + INDUSTRY**

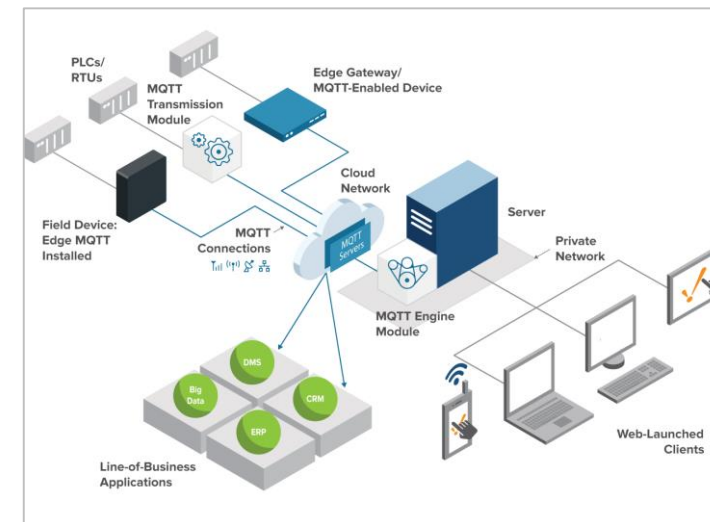
# TYPICAL SCENARIOS



Server All-in-one



Gateway connected to PLC (Backend).  
Frontend Gateway + Backend Gateway



Data Logging Box (SQL Bridge or Tag History,  
& Status Cache)

Edge Gateway: (Action to PLC capability, Data generation by Algorithm - Edge ML - )

Hub&Spoke: Local buffers + Data Hub

# FIRST ACTION: MAKE (AND SHARE) A PLAN

- The customer needs a plan to reach **his** results (**not your**)
- He is willing to spend, but not everyone agrees with him on HOW to spend
- It is partially aware of the actual problems its structure has

## Other «ITALIAN ASPECTS»:

- **Ci sono le parrocchie anche in Industria**
- **Essere agnostici «non fa figo»**
- **Un conto è risolvere un problema, un conto è entrare per anni nel suo sistema produttivo**

LET'S START THE SURVIVAL  
COURSE

## MAIN STEPS



- Don't talk about computing, cloud or anything else
- Make them understand what you can actually get and ASK if it is enough
- Don't talk about technologies or brands
- Set yourself as SERVICE OF, and not AT THE CENTER OF

# TOOLBOX-A: MEASUREMENT

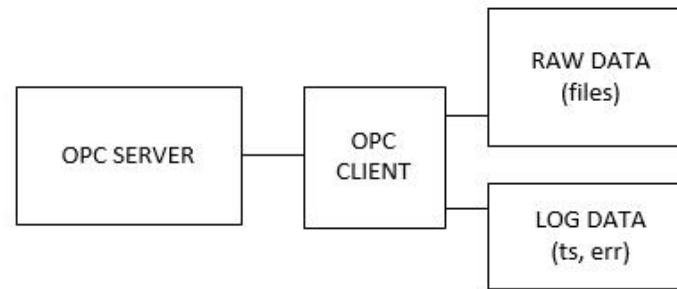
- Use Cloud and Edge to have a complete measurement
  - Understand how to get data from the machinery
  - Collect datablock or tags, get the data out of the plant, and re-simulate data pump in a message driven architecture:
    - Convert json
    - Analyze time series
    - Analyze RMS
- Compose a report with aggregations and evidences





# MEASURE DATA: EDGE FEATURES

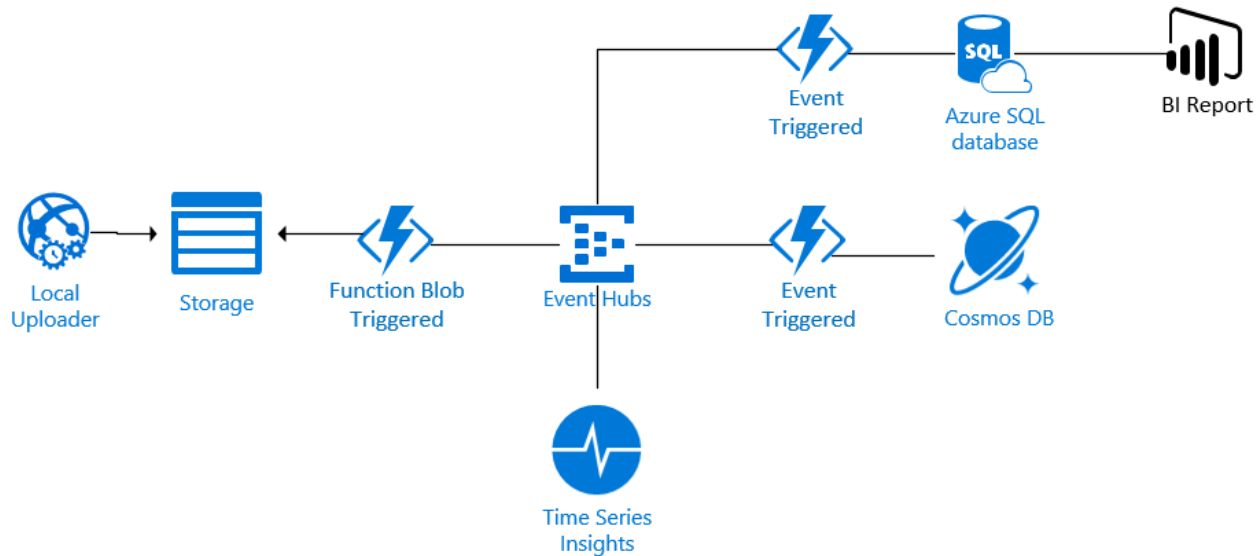
- Must be scriptable
- No conversion libs
- Near to the data - Storage SSD
- No Histeresys, only polling
- Remote management



**YOU CAN'T DO THAT WITH YOUR HOME PC**

1. Power Supply 24v
2. UPS 24v
3. OS LTS
4. Welding components
5. RTCodesys ready
6. X86 / ARM
7. OPC Server ready

# MEASURE DATA: CLOUD FEATURES



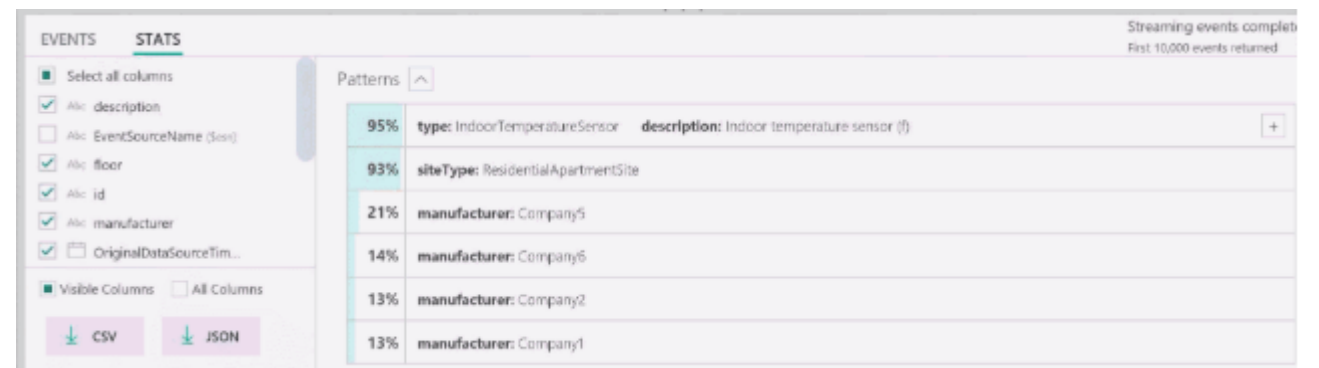
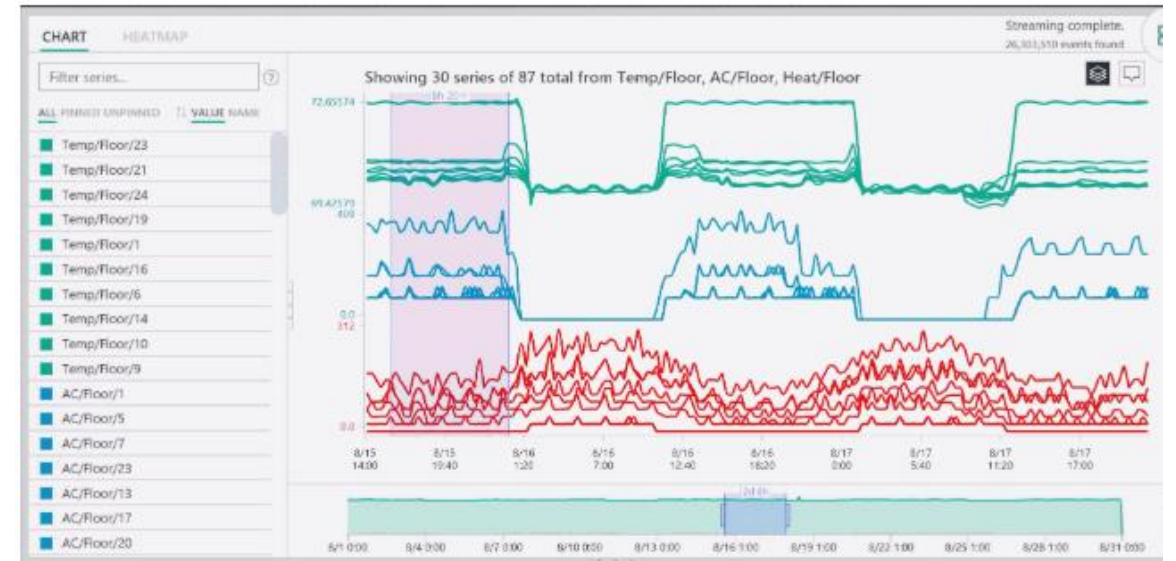
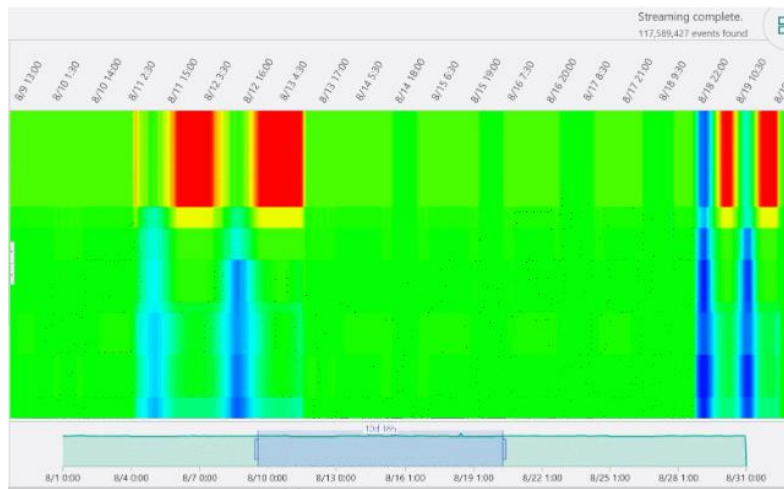
- Must be scriptable
- Conversion Lib must be the first step
- Payload decoration (PT, DT, DID)

## RECEIPE:

- SETUP: 30 minuts
- DEVELOPMENT: 8 hours
- ANALYSIS: 1 day
- REPORT: 1 day

# MEASURE DATA: REPORT FEATURES

- List of Variables
- Line chart variations
- Statistic view
- HEATMAP usage



# PAY ATTENTION TO DETAILS

- Could be useful:
  - Tag to JSON converter
  - Data Enricher
  - Data Confidence Tagger

☐ SELECT StartTime, EndTime, DATEDIFF(MILLISECOND, StartTime, EndTime) AS MetricValue  
FROM FT\_MACHINE\_MONITORING  
WHERE E0001 = 1 AND V0001 = 2 AND V0002 = 2 AND StartTime < '2019-06-11 10:10:30.233'

Timestamp	Source	Server	Message
12/06/2019 14:2...	Server Node		Could not connect to server: BadTimeout
12/06/2019 14:2...	DiscoveryWidget		Discovery FindServers on opc.tcp://localhost:4840 fail
12/06/2019 14:2...	UaExpert		UaExpert is ready to use.
12/06/2019 14:2...	UaExpert		Loaded GDS Plugin (Static Plugin).
12/06/2019 14:2...	UaExpert		Loaded Data Logger Plugin (Static Plugin).
12/06/2019 14:2...	UaExpert		Loaded Server Diagnostic Plugin (Static Plugin).
12/06/2019 14:2...	UaExpert		Loaded Image Viewer Plugin (Static Plugin).
12/06/2019 14:2...	UaExpert		Loaded UA Performance Plugin (Static Plugin).
12/06/2019 14:2...	UaExpert		Loaded UA Method Plugin (Static Plugin).

StartTime	EndTime	MetricValue
2019-06-11 10:01:30.233	2019-06-11 10:01:43.493	13260
2019-06-11 10:01:43.493	2019-06-11 10:01:44.523	1030
2019-06-11 10:01:44.523	2019-06-11 10:01:45.587	1063
2019-06-11 10:01:45.587	2019-06-11 10:01:46.633	1046
2019-06-11 10:01:46.633	2019-06-11 10:01:47.667	1033
2019-06-11 10:01:47.667	2019-06-11 10:01:48.703	1036
2019-06-11 10:01:48.703	2019-06-11 10:01:49.827	1123
2019-06-11 10:01:49.827	2019-06-11 10:01:51.047	1220
2019-06-11 10:01:51.047	2019-06-11 10:01:52.283	1236
2019-06-11 10:01:52.283	2019-06-11 10:01:53.563	1280
2019-06-11 10:01:53.563	2019-06-11 10:01:54.630	1066
2019-06-11 10:01:54.630	2019-06-11 10:01:55.597	966
2019-06-11 10:01:55.597	2019-06-11 10:01:56.613	1016
2019-06-11 10:01:56.613	2019-06-11 10:01:57.793	1180
2019-06-11 10:01:57.793	2019-06-11 10:01:58.840	1046
2019-06-11 10:01:58.840	2019-06-11 10:01:59.873	1033
2019-06-11 10:01:59.873	2019-06-11 10:02:00.910	1036
2019-06-11 10:02:00.910	2019-06-11 10:02:01.893	983

```
2019-06-11 12:10:59,209 [41] [ERROR] - OPC - Multiple read error. Ex: Error establishing a connection: BadRequestTimeout.
2019-06-11 12:10:59,303 [37] [DEBUG] - [Sentinel - Data acquisition service - sendingPointsTask] handled 0 in 0 ms; connection open: True;
2019-06-11 12:10:59,303 [4] [DEBUG] - [sendingSnapshotsTask] handled 0 in 0 ms; connection open: True; model open: True;
2019-06-11 12:10:59,725 [3] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:00,241 [41] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:00,319 [4] [DEBUG] - [sendingSnapshotsTask] handled 0 in 0 ms; connection open: True; model open: True;
2019-06-11 12:11:00,319 [5] [DEBUG] - [Sentinel - Data acquisition service - sendingPointsTask] handled 0 in 0 ms; connection open: True;
2019-06-11 12:11:00,742 [41] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:01,242 [37] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:01,320 [4] [DEBUG] - [Sentinel - Data acquisition service - sendingPointsTask] handled 0 in 0 ms; connection open: True;
2019-06-11 12:11:01,320 [5] [DEBUG] - [sendingSnapshotsTask] handled 0 in 0 ms; connection open: True; model open: True;
2019-06-11 12:11:01,757 [37] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:02,273 [5] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:02,335 [33] [DEBUG] - [Sentinel - Data acquisition service - sendingPointsTask] handled 0 in 0 ms; connection open: True;
2019-06-11 12:11:02,335 [4] [DEBUG] - [sendingSnapshotsTask] handled 0 in 0 ms; connection open: True; model open: True;
2019-06-11 12:11:02,773 [3] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:03,289 [37] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:03,351 [5] [DEBUG] - [sendingSnapshotsTask] handled 0 in 0 ms; connection open: True; model open: True;
2019-06-11 12:11:03,351 [41] [DEBUG] - [Sentinel - Data acquisition service - sendingPointsTask] handled 0 in 0 ms; connection open: True;
2019-06-11 12:11:03,790 [33] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
2019-06-11 12:11:04,305 [38] [ERROR] - OPC - Multiple read error. Ex: BadConnectionClosed.
```

```
2019-06-11 12:00:45,097 [5] [INFO ] - Connecting to opc ua: Services.Managers.Domain.Entities.OpcUa
2019-06-11 12:00:45,097 [5] [INFO ] - securityPolicy: Basic256
2019-06-11 12:00:45,097 [5] [INFO ] - messageSecurity: SignAndEncrypt
2019-06-11 12:00:45,097 [5] [DEBUG] - Client trying to connect...
2019-06-11 12:01:00,664 [5] [DEBUG] - OPC - Client Initialization error. Ex: BadRequestTimeout - Type:
2019-06-11 12:01:00,664 [5] [DEBUG] - OPC - Client: UAClient initialization failure. Started reconnection procedure.
2019-06-11 12:01:01,679 [5] [DEBUG] - Client trying to connect...
2019-06-11 12:01:01,679 [5] [DEBUG] - OPC - Started Disconnection procedure.
2019-06-11 12:01:16,718 [5] [DEBUG] - OPC - Client Initialization error. Ex: BadRequestTimeout - Type:
2019-06-11 12:01:16,718 [5] [DEBUG] - OPC - Client: UAClient initialization failure. Started reconnection procedure.
2019-06-11 12:01:17,720 [5] [DEBUG] - Client trying to connect...
2019-06-11 12:01:17,720 [5] [DEBUG] - OPC - Started Disconnection procedure.
2019-06-11 12:01:26,850 [5] [DEBUG] - Endpoint selected: http://opcfoundation.org/UA/SecurityPolicy#Basic256 SignAndEncrypt
2019-06-11 12:01:26,899 [5] [DEBUG] - OPC - Certificate accepted: CN=IPC_70715, E=opcua@siemens.com, OU=Industry, O=Siemens
2019-06-11 12:01:28,192 [5] [DEBUG] - OPC - Created new client: UAClient. Address: opc.tcp://192.168.214.241:4840
2019-06-11 12:01:29,193 [5] [INFO ] - Client connected!
```

Once upon a time there was a big drill and a PLC



# TOOLBOX-B: DATA MAPPING & DATA FLOW

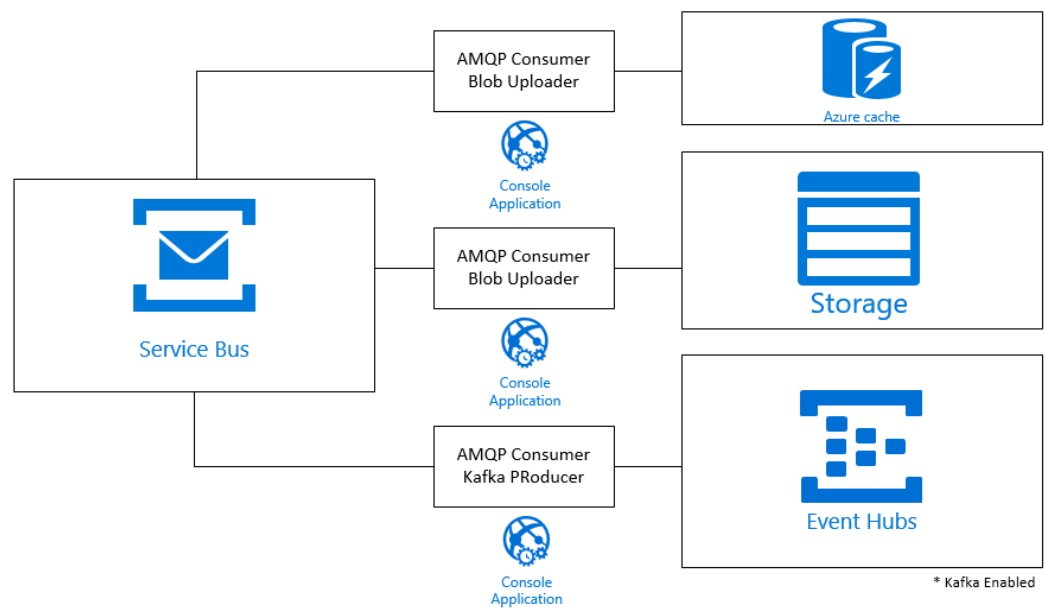
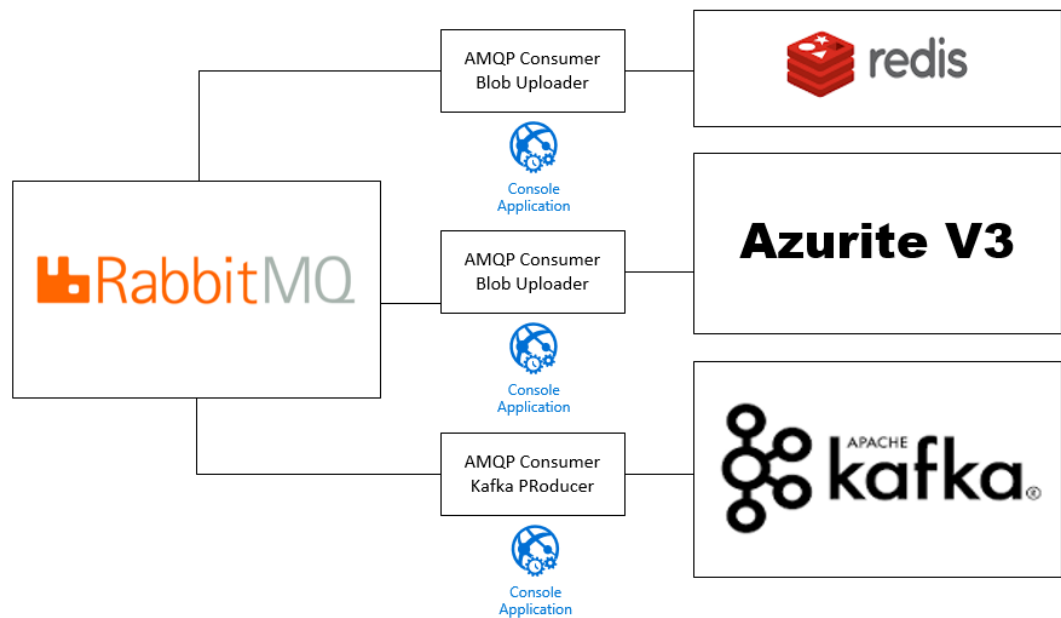
- Convert the VariableList to Protocol based schema
- Enrich with plant static dimensions
- Divide data into «FLOWS» :
  - Based on variation
  - Based on updated snapshot
- Divide data considering «DATA.TEMPERATURE» (lambda 3F)
- Divide data based on retention time



## DATA MAPPING & DATA FLOW

[illegible]

# LFS ARCHITECTURE: «3 PIPES FLOW»



# DATA MAP & DATA FLOW: EDGE CONSIDERATIONS

1. Understand protocols and uniform data flow FIRST
2. Understand frequency data
3. Understand network load
4. Understand data origin (SCADA, Industrial PC, OPC SERVER, ..Direct S7.. etc)

Pay attention to:

- Disconnections: detect them and tag period of time that are globally inconsistent
- Network: BSOD are the daily problem if you use virtual network / Docker at the edge



# DATA MAP & DATA FLOW: EDGE CONSIDERATIONS

- Solution: good EXCEL with frequency report, throughput calculation and network schema
- Fog and Cloud Are not Mandatory. A Good edge layer with alerts and File based aggregations could be sufficient

Network  
Schema

Good  
Excel

Throughput  
calculation

Payload Families,  
based on  
Temperature

Load Calculation

# CHECKPOINT NO. I

- Intent Declaration, and ask for official confirmation
- Proposition in terms of Apps and Users
- And Remember Mario's needs... **non i  
bisogni del cloud, del design, o della  
«moda tecnologica» in genere.**



# PAY ATTENTION TO THE INTENT DECLARATION

**PROBLEM 1:** Where is the right place to normalize data.

- Byte Stream, Json Stream? Depends on Frequency, network, RT needs

**PROBLEM 2:** Mimic to be used

- Realtime, batch. Organize thinking to the REAL goal

**ONLY ONE IMPORTANT THING:**

- Spend your time in order to **DEVELOP THE BEST SIMULATOR YOU CAN**
- Certe cose vanno fatte SOLO in fabbrica. Scordiamoci di avere «scontata» una soluzione cloud-edge per i prossimi 5 anni. Le uscite dal cliente VANNO FATTE.

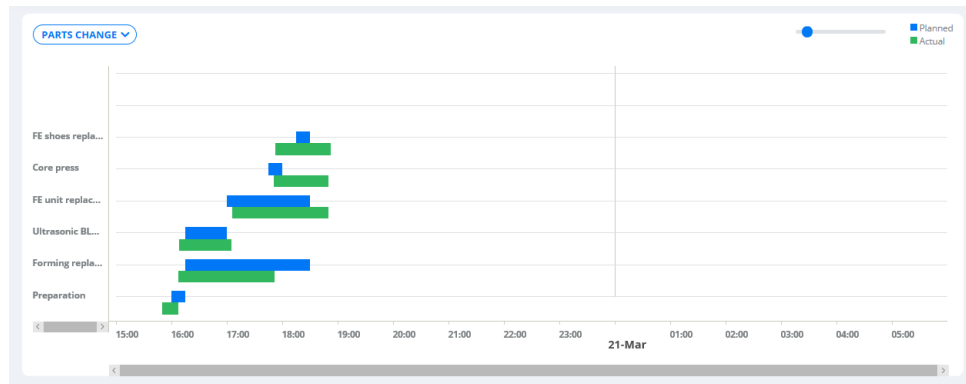
# FOCUS ON FREQUENCY AND MIMIC

Divide data into many flows, depending on Application Goals:

- PRODUCTION MONITORING
- PROCESS CONTROL
- CONTROL PLAN
- MAINTENANCE PLAN
- CHANGE OVER PLANIFICATION
- WORK ORDER PROGRESSION
- CORRELATION ANALYSIS
- STOPPAGES ANALYSIS
- RAW MATERIAL OPTIMIZATION

# EXAMPLE

## Change Sequencing

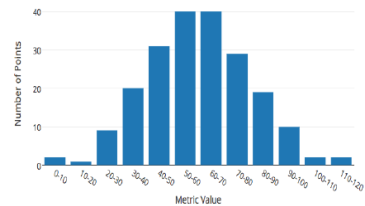


## Carta XR

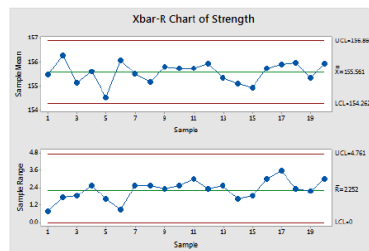
Quality parameter

04/24/2018 - 04/24/2018

Si seleziona un parametro dal menu a tendina misurato dal Sistema di visione ed un periodo temporale



Istogramma con la distribuzione delle misure (o le mediane)

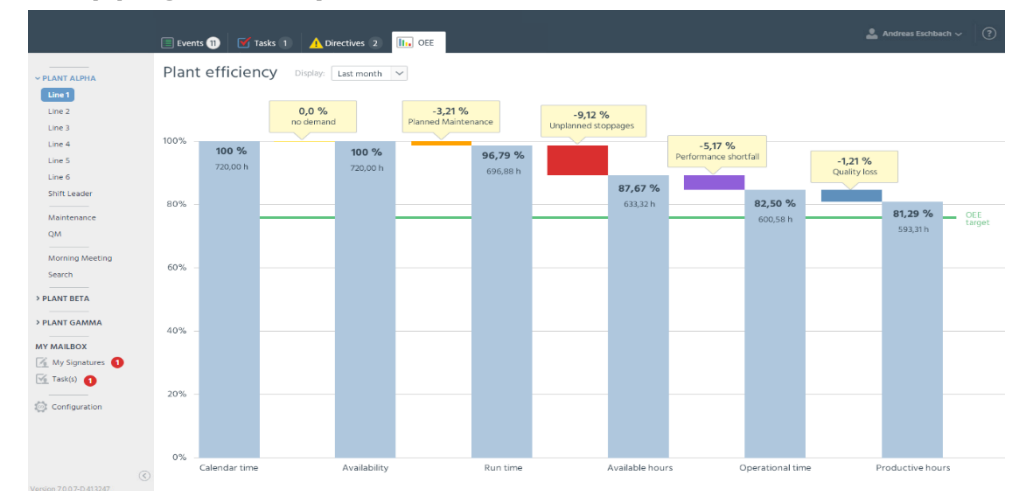


Andamento di Mediana e Range nel tempo

## Audit sui Control Plan

#Rif.	CHECK DIMENSION	TARGET	USL	USL	tolerance	Vision system			
FINISHED PRODUCT		Drawing Tolerance				Average	Standard deviation	Cp	Cpk
D0b	Lunghezza prodotto finito Finished product length	430.0	425.0	435.0	5				
D0c	Larghezza Prodotto Finito - LD Finished Product width - Back	315.0	305.0	325.0	10				
CORE									
DSa	Lunghezza Tampone Core length	340.0	335.0	345.0	5				
DSb	Larghezza Tampone - LA Core width - front	90.0	85.0	95.0	5				
DSf	Posizione CD tampone a backsheet (LD-LT) centraggio Core CD position to backsheet - (DS-DS) centred	0.0	-6.0	6.0	6				

## Stoppages Analysis



## MODE DEFINITION («FREQ&TEMP»)

- PRODUCTION MONITORING => STREAMING, LOW FREQUENCY
- PROCESS CONTROL => STREAMING, HIGH FREQUENCY
- CONTROL PLAN =>
- MAINTENANCE PLAN => BATCH, LOW LOAD
- CHANGE OVER PLANIFICATION => BATCH, LOW LOAD + STREAMING + LOW FREQ
- WORK ORDER PROGRESSION => STREAMING, LOW FREQUENCY
- CORRELATION ANALYSIS => BATCH, HIGH LOAD
- STOPPAGES ANALYSIS => BATCH, LOW LOAD
- RAW MATERIAL OPTIMIZATION => BATCH, LOW LOAD

# IDENTIFY ACTORS & ARCHITECTURES

- Maintenance squad
  - Control room
  - Operator
  - COO
- 
- Can we solve their problems with one architecture?? NO.
  - Is FOG the unique workplace? NO.

# TECHNICAL ELEMENTS FOR THE PERFECT EDGE

- **MinIO** is High Performance Object Storage
- **InfluxDB** is Perfect for TimeSeries Buffering
- **Azurite** is Perfect to have 3 Structure in One
- **FunctionRuntime** is perfect to have State-Machine
- **SQLite** is perfect to store registry, and transport configuration
- **Rabbit** is perfect to process/buffer/Divide data
- **Redis** is perfect to have a Repository with Last
  - **REDIS USAGE** in
    - Pattern: Reliable queue
    - Pattern: Circular list
    - Pattern: Status Snapshot K,V
    - Pattern: Publish Subscribe
  - **RABBIT USAGE** in
    - Exchange to QUEUE for Lambda cold/hot pushing
    - Exchange to TOPIC multi Executor pattern-matching
    - Exchange to QUEUE for Enriching data with Plant/Machine Metadata Cache(Redis K,V)



# BUT THE KEY FACTOR... IS TO HAVE CONFIDENCE WITH HIM

- Only Proc. Manager can define
  - OEE meaning in THAT context
  - What is useful to measure (stoppages, process vars)
  - What are the Formulas that must be used (example: XR Chart)
  - What have we to correlate
  - Which are the additive sensors that must be implemented before go on, in order to trigger checkpoint / tracking during process

**He Knows!!!**



# WITH THE PROC. MANAGER BY YOUR SIDE, YOU WILL..

Push the Stop Cause Button when your line stops  
→ Automatically registers stop causes on the cloud application.

## **DOWNTIME ANALYSIS**

Check-in gets signals from PLC automatically  
→ Visualises the machine status on the graph and chart.

## **MONITORING ON A SMARTPHONE**

Check-in gets signals from andon lights  
→ Sends an SMS to operators when the light turns red.

## **PUSH MESSAGES ON A MOBILE PHONE**

Check-in gets signals from stand-alone machines  
→ Monitor your machine status from anywhere.

## **ON/OFF STATUS MONITORING**

Check-in gets signals from movement at manual stations  
→ Digit manual processes and visualize progresses.

## **PROGRESS MONITORING**

Push the Issue Log Button when your line stops  
→ Automatically registers issues with time stamps on the cloud application.

## **ISSUE LOGGING**

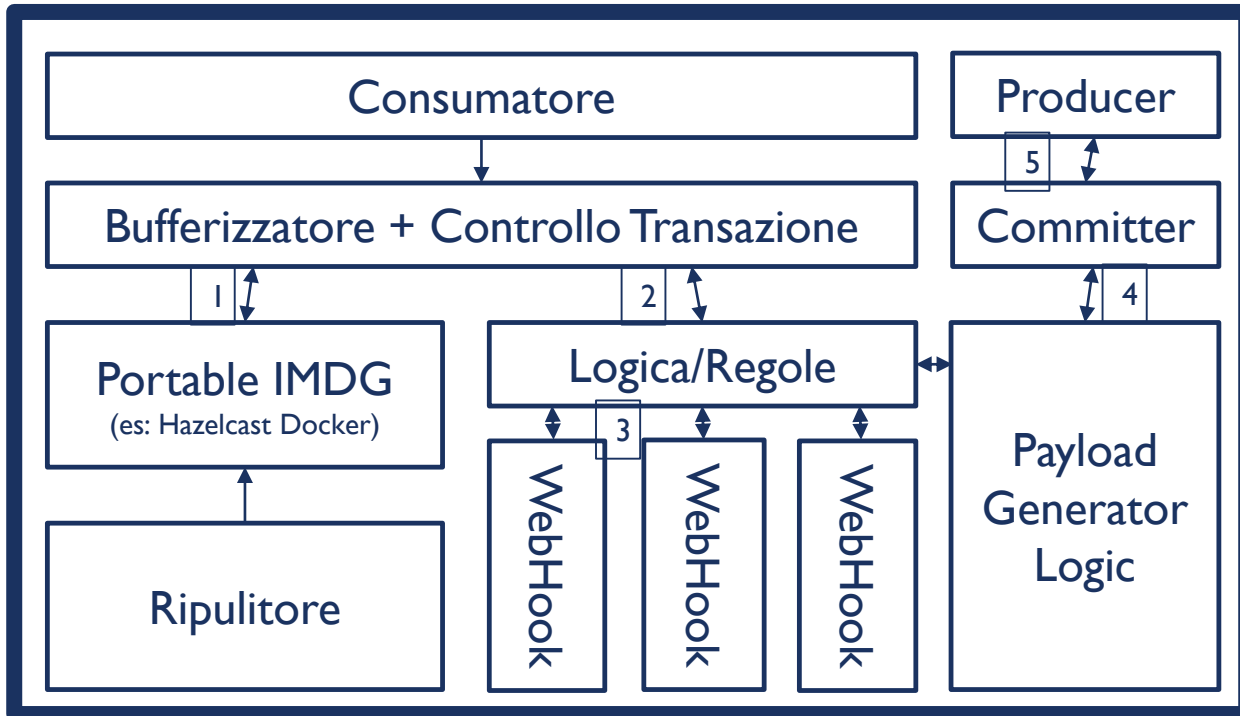


# TOOLBOX-C: INGESTION AND PROCESSING

- Go on with Architecture, using cloud to simulate 3Pipes
  - Develop Cold path (based on File e topic ad-hoc) with TSDB import
  - Analyze and generate new payloads (use Buffered Queue, Status Caches, to retrieve Start/stop payloads, tracking corr\_id attribute in every pipe)
  - Analyze rules and develop engines to generate alert using threshold
  - Create command flow, back to the edge



# CONSUME IS NOT EASY..



React with DLQ and Command to install RepairTools  
Decouple cleaner from users  
Use memory, not disk or network  
Use React hooking  
Use configurable Logic  
Use Bulks

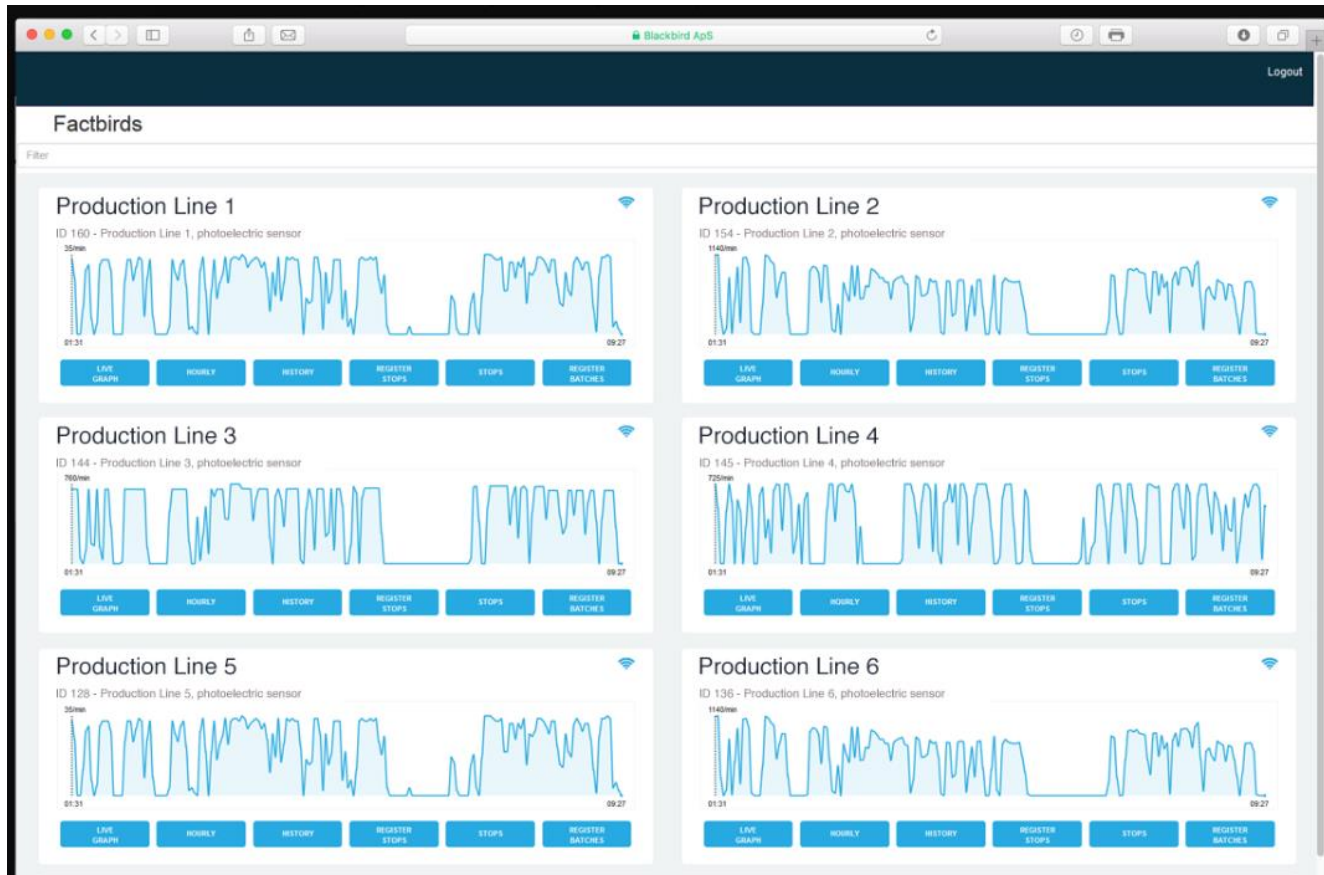
Admit Error  
Trace Error zones

# TOOLBOX-D: CHART... CHART EVERYTIME

- Use a common library, and make standard lib to generate chart types:
  - D3 / c3 / Plotly / ChartJS / ecc
  - C# / NodeJS / Python / Php / Java / ecc
- PowerBI / others are not for the onPremise
- Use transformer from data to Chart oriented Json

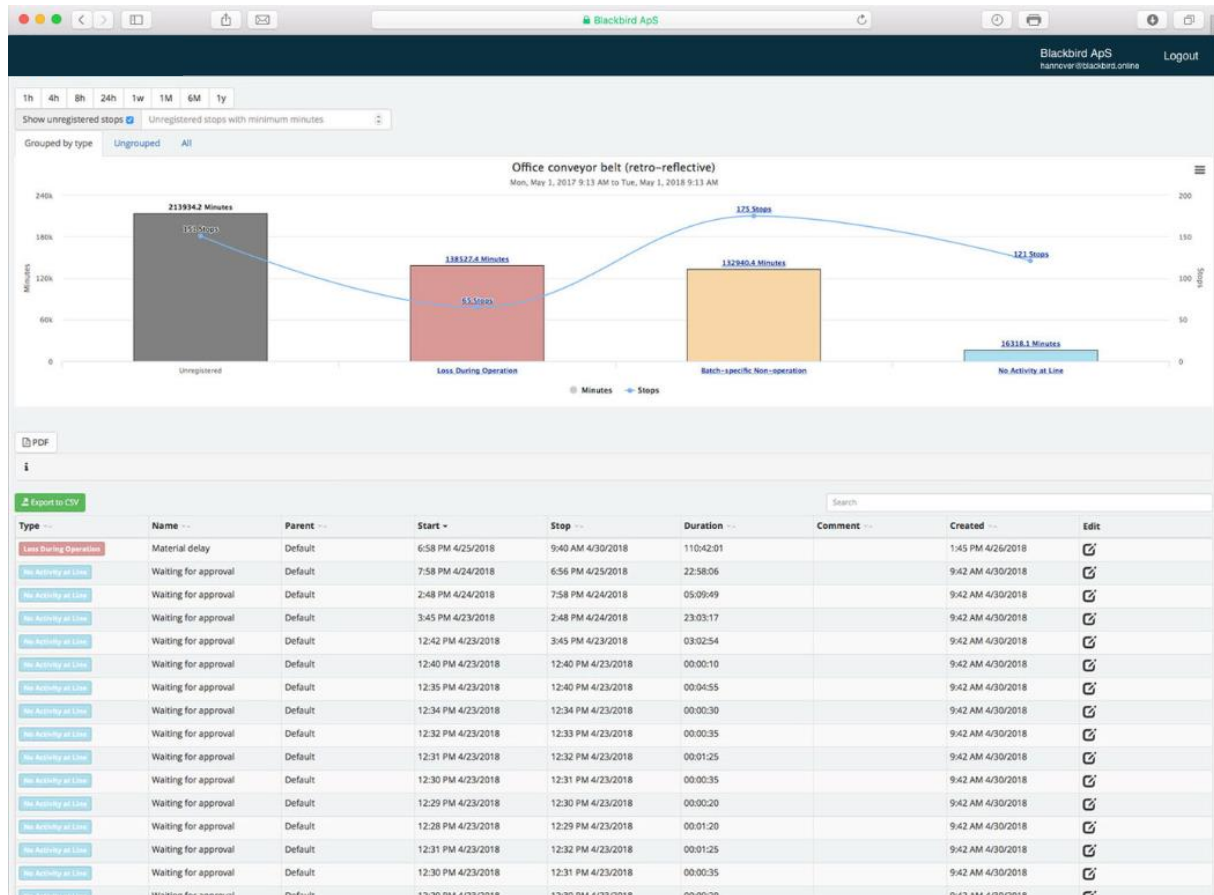


# SOME EXAMPLE



- IMPORTANT THINGS ABOUT CHARTING
  - View the actual status
  - Compare period of time (2 charts, never only 1)
  - Save filters used during searches
  - Enrich data with information, annotations or manual attributes in order to UNDERSTAND (ex: STOPPAGES ANALYSIS)

# SOME EXAMPLE



- Usage statistics must be correlated to Shifts, Teams, Production Order, Receipt. You will find:
  - any lack of training of operating personnel
  - any missing information derived from the MES
  - Possible performance deficiencies in terms of format change

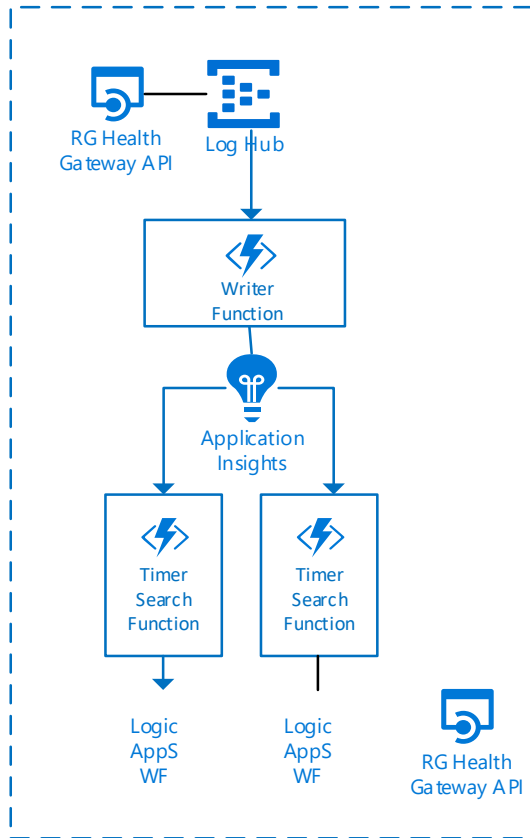
# TOOLBOX E: DATA QUALITY EVALUATION

- Go on with solution design:
  - Tag, with batch process, all data zones (grey zones) that are globally «not good, or partial ingested»
    - Opc Server Disconnection
    - Edge restart
    - Network failure
  - Give evidence in every dashboard about data quality with an overall % in terms of Good Data
  - Give evidence is there is an anomaly in the system that increase the number of grey data zones





# EXAMPLE: AZURE REFERENCE ARCHITECTURE



- Entry point (Rest API)
- Stream Logger (EventHub, Kafka)
- Consumer (Console application)
- Log Data Ingestor (Application Insight)
- Timer based Function
- Logic Apps (Workflow)

# NOW THE QUESTION IS ....AND THE CLOUD??

- If the Proc. Manager is scared, it doesn't matter. Just make him think and he will be your first sponsor.



# WHICH ARE THE REAL QUESTIONS TO MAKE TO THE SPONSOR

- Which is the data that must be «ON PREMISE»
- Which are the Key users to be focus and transformed into sponsor
- Which are the already present players?

And the solution is:

- 1) Create and export SIMPLE databases with only the data that is plotted (store the sum)
- 2) Do not send process data, but trigger a process of queuing requests and asynchronous export.
- 3) The important thing is not to generate inbound traffic, but only outbound aggregated data from a side DB "without secrets".

# WHICH IS THE CLOUD ROLES

- Simulaton desk
  - Throughput
  - Analytics
  - Inference
- Information Bus
  - To Customers
  - To Scientists
- Extension Workplace
  - ML Training Workbench
  - Module Repository



Once you've made your  
complete simulated  
solution... now you can





START THE ADVENTURE WITH PROUD & GLORY