



La piattaforma Bigdata di Sparkle e l'evoluzione IA

Linux Day 2023

Sparkle – The company

Mission

Sparkle is a TIM-owned company that operates in a global context offering connectivity solutions and ICT services to meet the needs of multinational companies, internet service providers, OTTs , Media and Content Player, fixed and mobile phone application providers and carriers.

Sparkle owns and operates a proprietary network of over 600,000 km of land and sea cables with a widespread presence in the Mediterranean basin, Europe and America. With a direct presence in 32 countries and commercial coverage on a global scale, Sparkle can count on a workforce practically distributed all over the world.

SPARKLE



2023

CC Global Awards
Best Data/Capacity Provider



2022

CC Global Awards
Best Digital Transformation Programme



2023

CC Global Awards
Best Digital Advertising Operator



2022

CC Global Awards
Best Data/Capacity Provider

Presence WW



Sparkle delivers the most sophisticated communications with the highest quality and reliability through an unmatched state-of-the-art global backbone of terrestrial fiber and international submarine cable.



2022

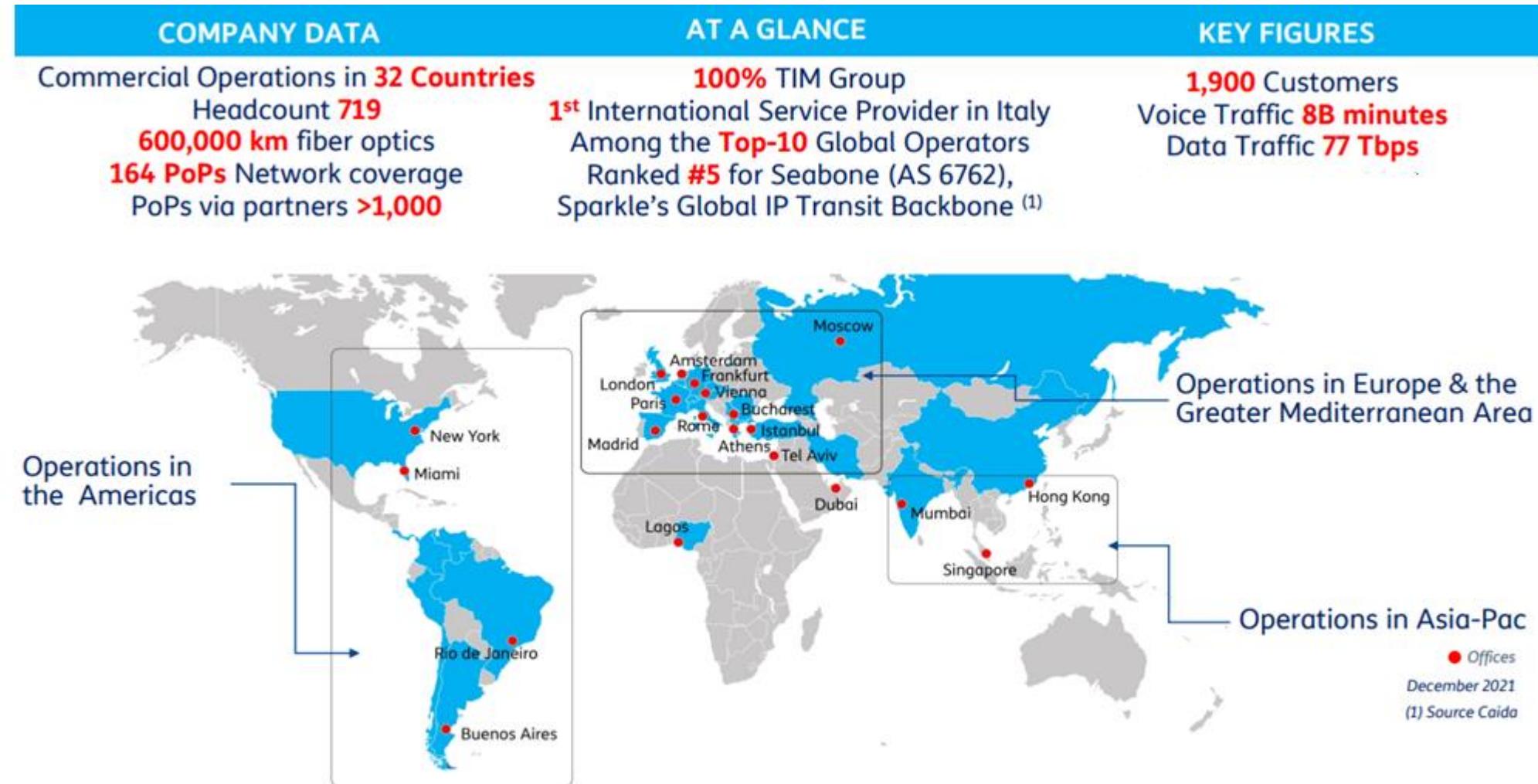
Global Carrier Awards
Best Subsea Innovation



2021

World Communication Awards
Customer Experience

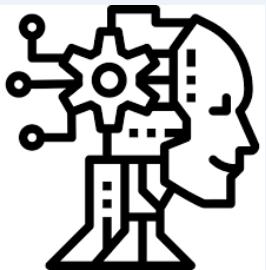
Sparkle – Main Indicators



AI is general purpose technology

AI

Techniques that allow to emulate human behavior

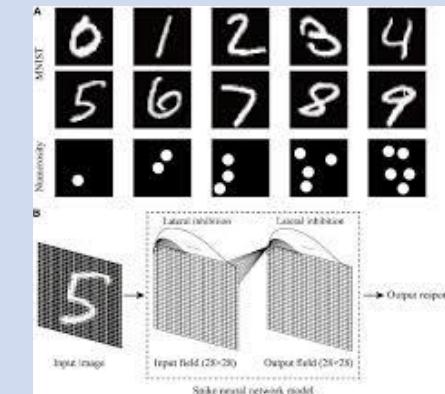


Machine Learning

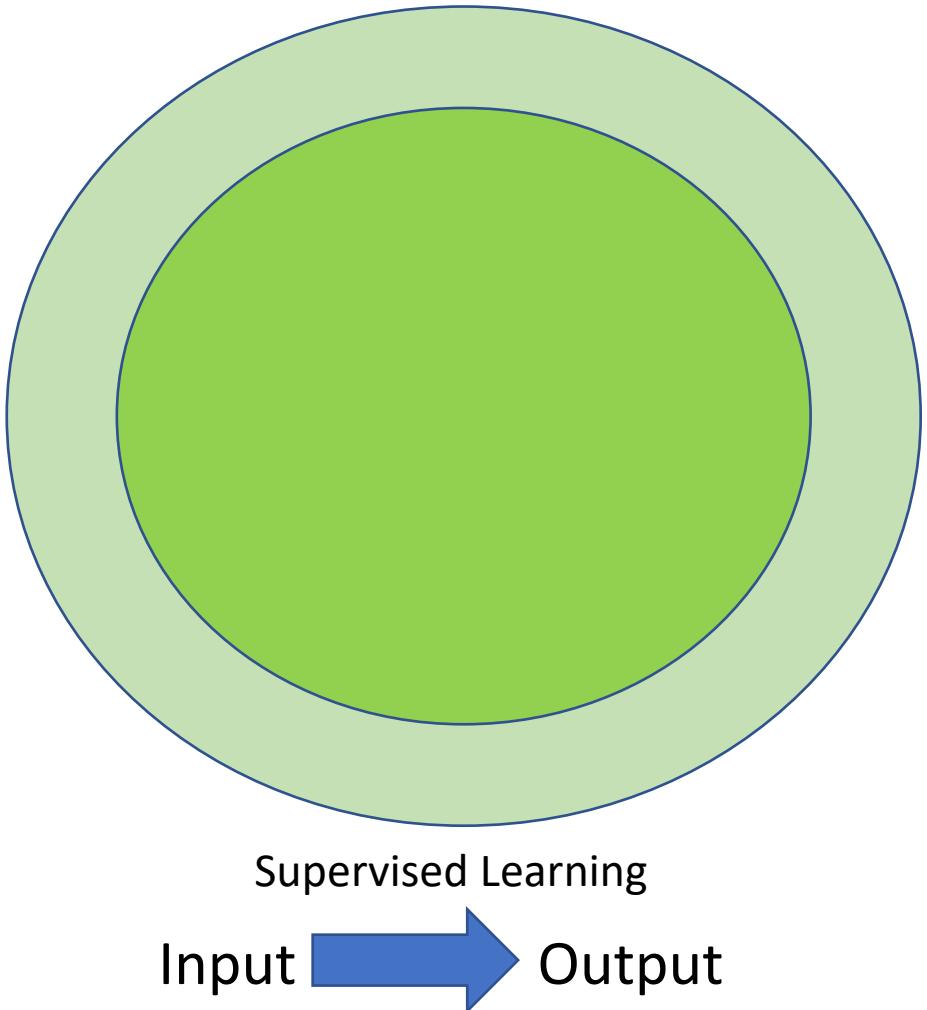
Machines that have the ability to learn without being explicitly programmed

Deep Learning

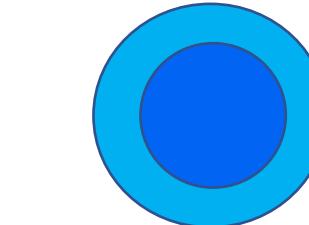
Extraction of pattern from data using Neural Networks



Value from AI Technologies: -> 3 years



Generative AI



Unsupervised Learning

Today the most relevant part of the value of AI project is concentrated on Ads and Web Search solutions



Reinforced Learning

Supervised vs Generative AI

Input (A)	Output (B)	Application
Email	Spam/No Spam	Spam filtering
Image, Radar Info	Position of other cars	Self driving car
Restaurant reviews	Positive/Negative	Reputation
...		

Generative AI is built by using supervised learning to repeatedly predict next word

Adoro mangiare (input/prompt)

**risotto con gli asparagi
l'arrosto preparato con gli amici
la pasta con le sarde preparata da mia madre
fuori con gli amici**

output

Input (A)	Output (B)
Adoro mangiare	risotto
Adoro mangiare risotto	con
Adoro mangiare risotto con	asparagi
...	

Generative AI

DALL-E History Collections

Edit the detailed description

Foto di un astronauta che cavalca un cavallo rosso

Surprise me Upload →| Generate



Default (GPT-3.5)

< 2 / 2 >  Scrivi un programma che effettua il training di una rete neurale con Tensor Flow

Certamente! Ecco un esempio di programma Python che utilizza TensorFlow per allenare una rete neurale per il riconoscimento di cifre scritte a mano utilizzando il famoso dataset MNIST:

```
python
import tensorflow as tf
from tensorflow import keras

# Carica il dataset MNIST
mnist = keras.datasets.mnist
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

# Normalizza le immagini
train_images, test_images = train_images / 255.0, test_images / 255.0

# Crea il modello della rete neurale
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dropout(0.2),
    keras.layers.Dense(10)
])

# Definisci la funzione di perdita e l'ottimizzatore
loss_fn = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = 'adam'

# Compila il modello
model.compile(optimizer=optimizer,
              loss=loss_fn,
              metrics=['accuracy'])

# Allenamento della rete neurale
model.fit(train_images, train_labels, epochs=5)

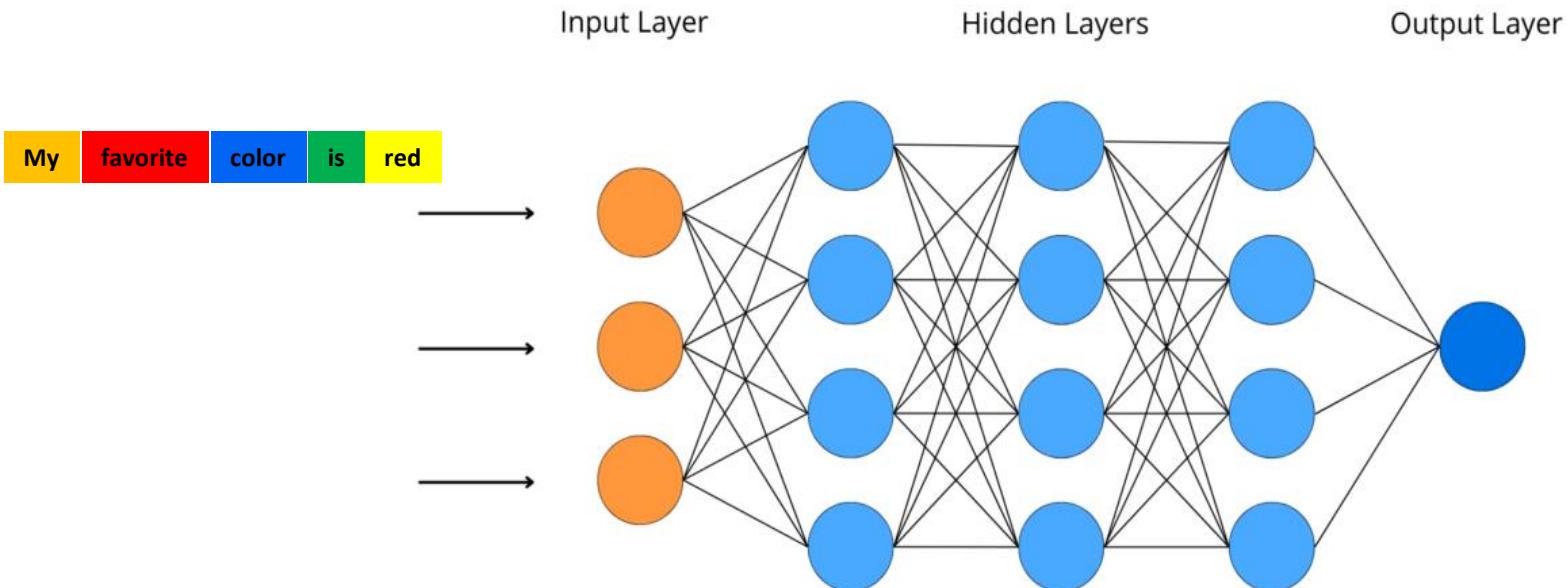
# Valutazione del modello
test_loss, test_accuracy = model.evaluate(test_images, test_labels, verbose=0)
```

Regenerate

Send a message

Large Language Models (LLM)

Text Documents (tokens)



3 main kinds of LLM:

- Generic (next token);
- Instruction tuned;
- Dialog tuned;

Reinforce learning:
Es Proximal Policy Optimization (PPO)

LLM can be **pre-trained** and fine-tuned for specific purposes;

LLM are trained to solve common language problems: text classification, question answering, document summarization, text generation, ...;

... then tailored to solve specific problems in different fields: Finance, entertainments, ...using a relatively small fields and datasets;

Benefit of LLM:

1. a single model can be used for different tasks;
2. fine-tune process requires minimal field data;
3. the performance is continuously growing with more data and parameters.

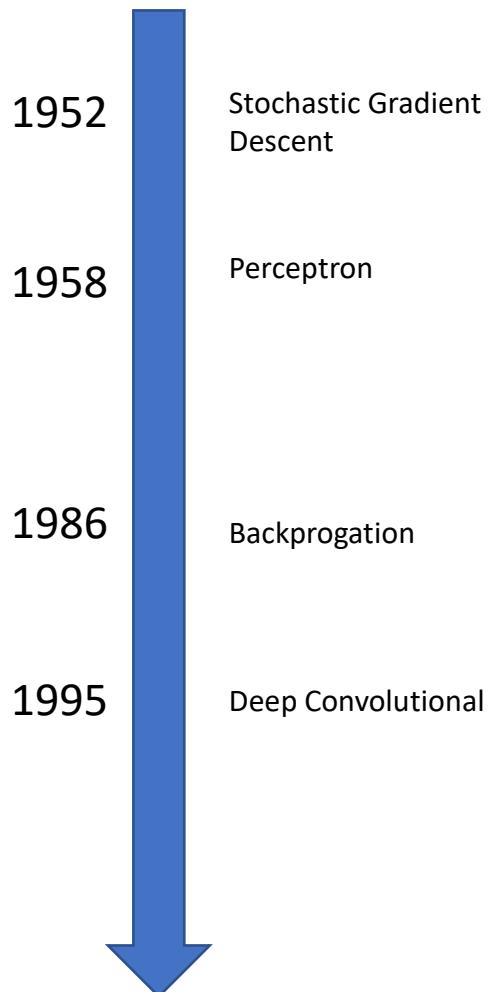
LLM projects:

- ✓ LaMDA;
- ✓ PALM (Bard);
- ✓ GPT-3.5/4 (175 B trainable parameters, 96 Layers; 500 B of tokens);
- ✓

AI Risks/Opportunities

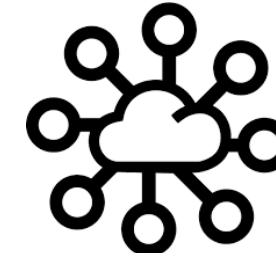
- ✓ AI (do anything a human can do) is still decades away;
- ✓ AI develops gradually and the «hard take off» scenario is not realistic;
- ✓ AI is an important piece of solution to the real existential risks to humanity (next pandemic, climate change,...);
- ✓ AI problems: bias, fairness, accuracy....but the technology improve quickly;
- ✓ AI will disrupt many occupations.

The rediscovery of Neural Networks



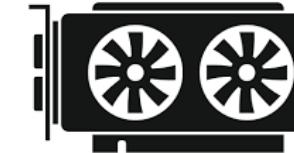
1. Big Data

- ✓ Larger datasets
- ✓ Easier collection and storage
- ✓ Centralized governance



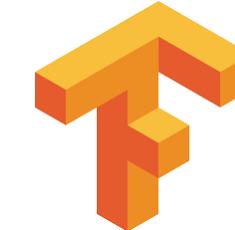
2. Hardware

- ✓ GPUs
- ✓ Parallel architectures



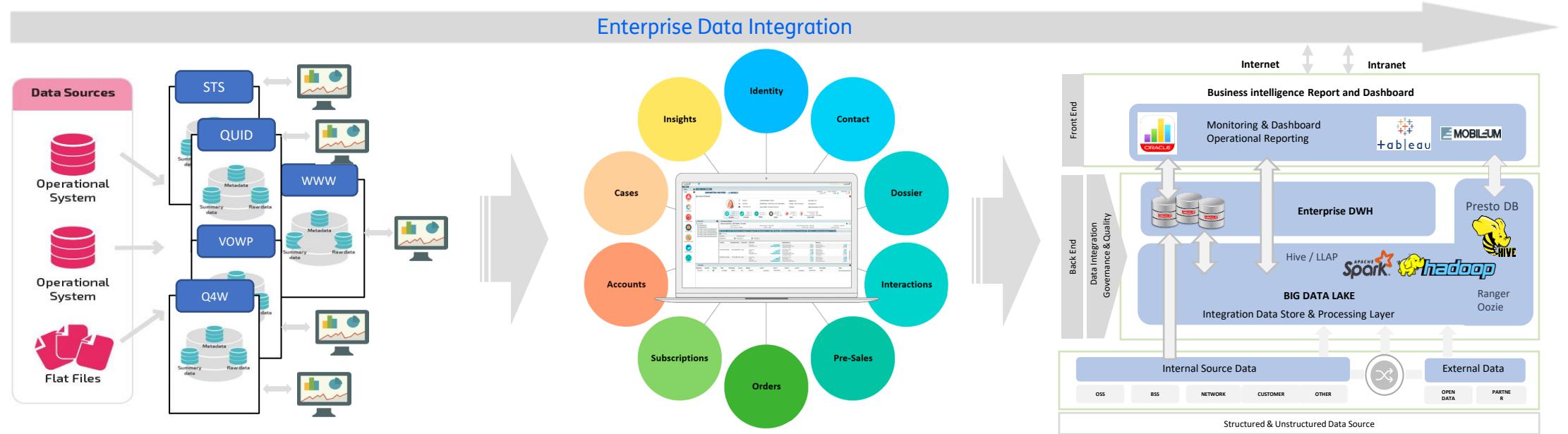
3. Software

- ✓ Improved techniques
- ✓ New modes
- ✓ Toolboxes



From DWH to Big Data

The value of the evolution of Big Data has been **the unification of the data of the entire corporate group**, which is geographically distributed throughout the world and has various Legal Entities.



In the past decades Sparkle developed different DWHs for separate data analysis:

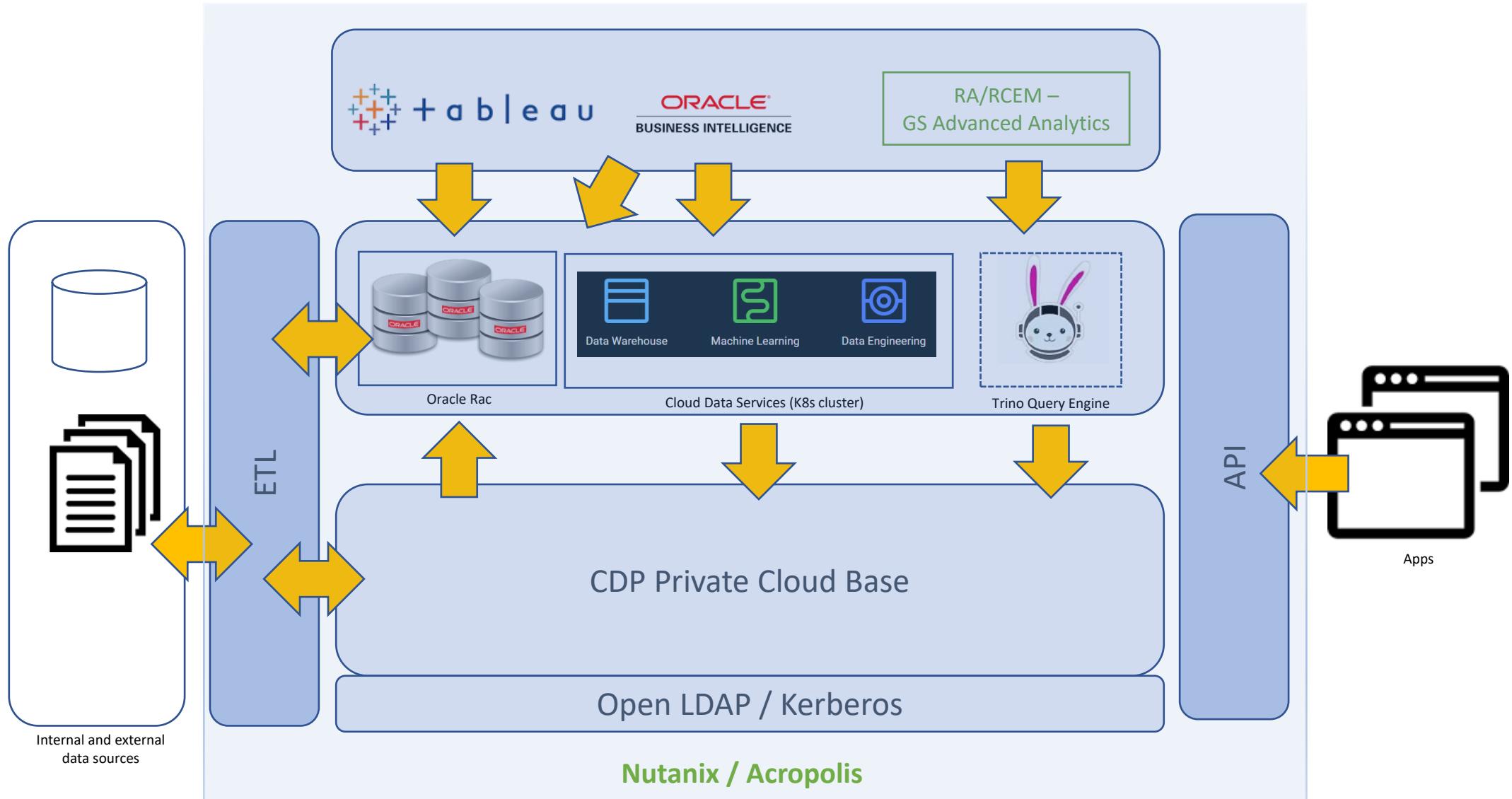
- STS - for elementary Voice Traffic
- WWW - for Voice and SMS aggregated and rated traffic
- QUID - for Voice and IP Quality

More recently the project One Customer View (OCV) realized a new **unitary vision for customer data**, based on the gathering from multiple sources and introducing the presentation on mobile devices (MobApp)

Big Data has been introduced to go beyond the single DWH and move towards the Enterprise Data Integration. Different streams have been defined in order to address the needs of different processes:

- Economics Performance & Predictions
- Products Dashboards
- Traffic and Quality Analysis
- Process Monitoring
- Proactive Monitoring

Big Data – Data Platform on Premises



Big Data – Data Platform on Premises / Open Sources

Data Storage



Data Ingestion & Streaming



Spring
Boot

Data Processing



Zeppelin

API



Spring

Data Governance



Atlas

DWH

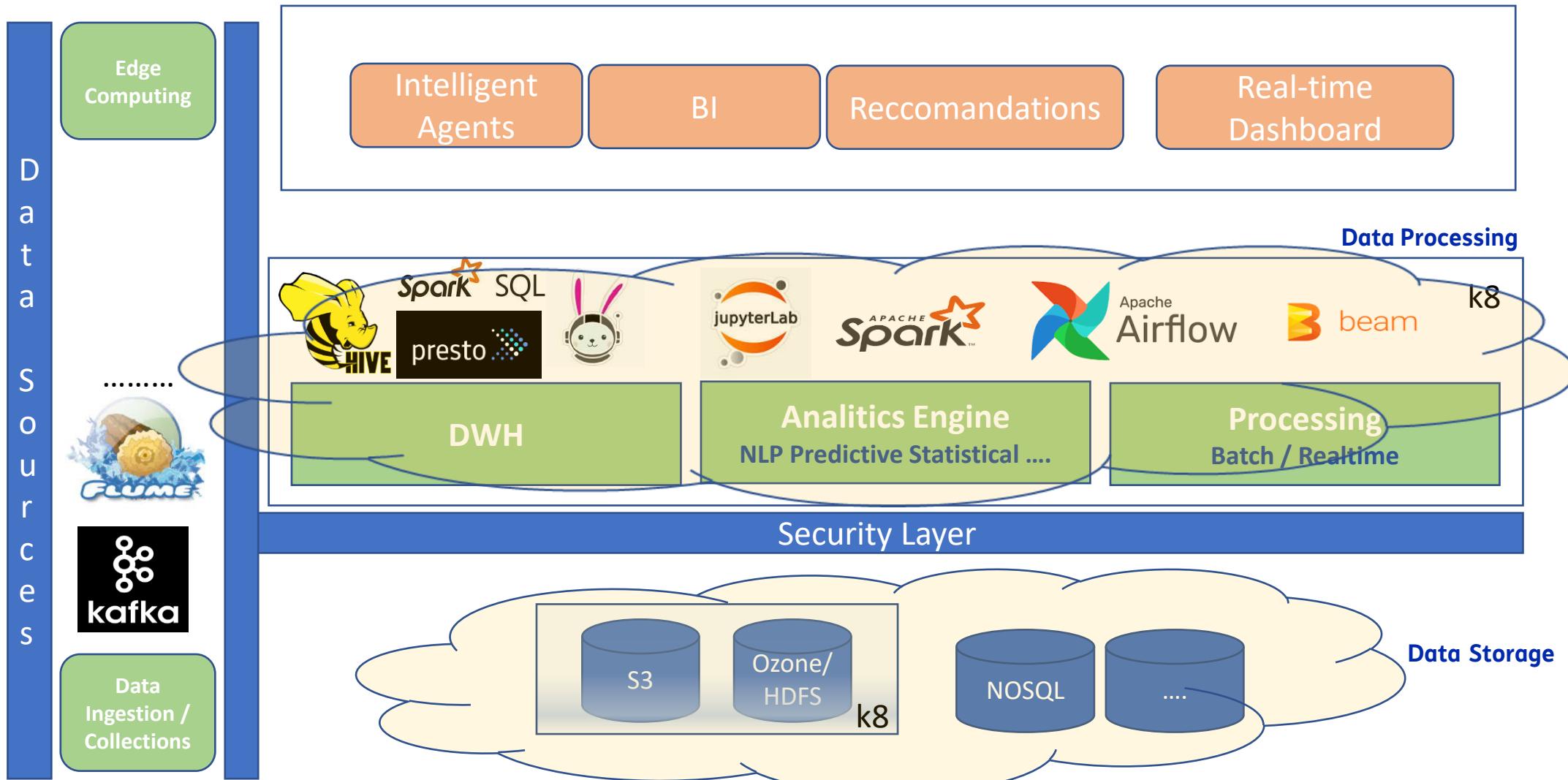


Workload Management

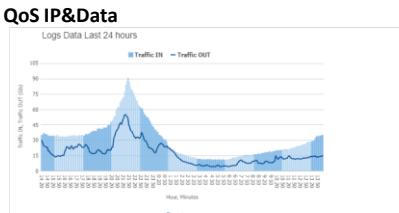
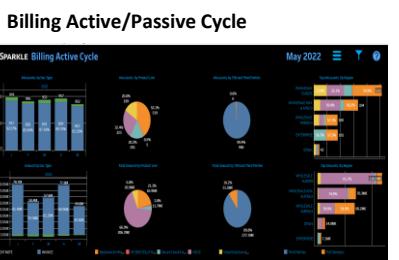
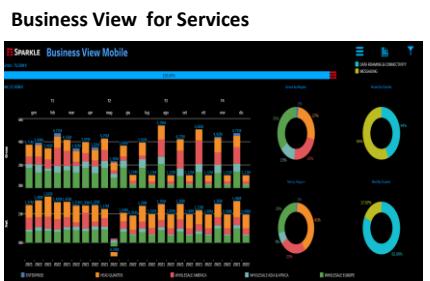
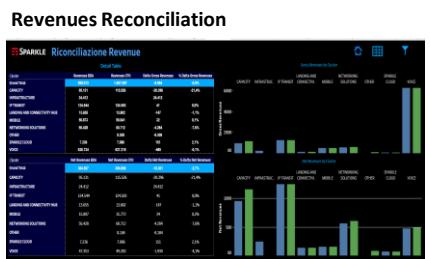


Big Data – Open source data platform

Data Visualization



Big Data – Use cases implemented



Name - Bundle	Service ID	Bandwidth	In	Out	UPL%																		
milano10- et-0105.0	TIS-0000001941	100	0.772	0.658	1	0.034	0.535	1	0.975	0.784	2	0.609	0.997	1	0.616	0.685	1	0.215	0.095	0			
milano31- et-0115.0	TIS-0000005416	100	0.696	1.287	1	0.376	3.666	4	2.831	3.614	22	0.169	0.992	1	0.201	0.851	1	0.219	0.314	1			
milano31- et-0115.0	TIS-0000004924	100	0.644	0.990	6	7.967	2.269	8	8.377	2.350	8	0.126	2.461	10	5.002	0.887	6	5.867	0.829	6	5.877	0.693	6
milano00- et-5115.0	TIS-00000063950	100	0.004	0.333	0	0.460	0.604	1	2.367	0.754	21	0.085	0.255	0	0.175	0.224	1	0.173	0.259	1			
milano10- ae16		100	0.072	0.698	1	0.367	2.179	2	2.902	2.993	21	0.085	0.962	1	0.188	0.905	1	0.194	0.368	1			
milano10- ae2		100	0.765	0.191	1	0.043	0.480	1	1.417	0.716	1	0.471	0.718	1	0.624	0.997	1	0.633	0.086	1	0.210	0.093	0
roma54- ae20		400	0.170	2.092	1	0.210	2.115	1	0.447	2.966	1	0.476	3.296	1	0.137	1.342	0	0.137	1.374	0			

```

File Edit View Run Kernel Tabs Settings Help
+ X C Code
Filter files by name
Name Last Modified
PERON A... 2 years ago
ProfitC... 2 years ago
ROUSSO... 2 years ago
SalesPerf... 2 years ago
SalesPerf... 2 years ago
SparkleC... 2 years ago
SpkCusto... 7 months ago
SpkCusto... 2 years ago
strategia... 2 years ago
Untitled... 2 years ago
Untitled... 2 years ago
region ref="EU"

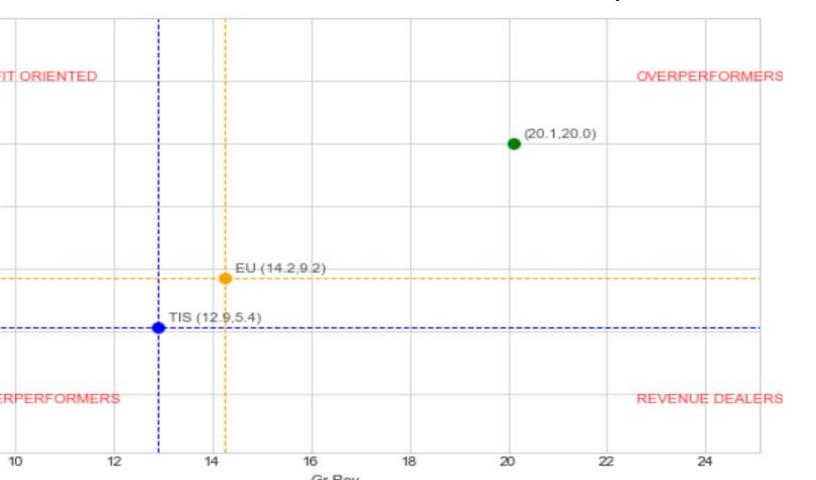
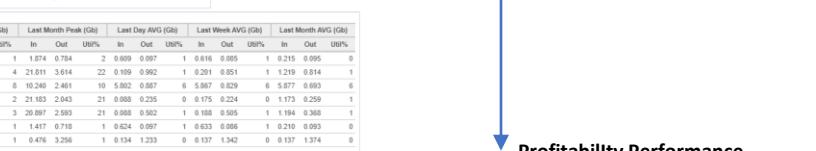
[3]: net_tis_curr=df.curr['NET'].sum()
print("NET TIS curr:",net_tis_curr)
net_tis_prev=df.prev['NET'].sum()
print("NET TIS prev:",net_tis_prev)
net_var_tis=(net_tis_curr-net_tis_prev)/net_tis_prev*100;
print("VAR % NET TIS:",net_var_tis)

NET TIS curr= 20761338.579371005
NET TIS prev= 22000309.019606
VAR % NET TIS= -5.631604715783146

[4]: df.curr_am_notnull=df.curr['Account Manager'].notnull();
net_am=df.curr_am_notnull['NET'].sum();
am_pd.unique(df.curr_am_notnull['Account Manager'])
net_avg_am=net_am/len(am)/1000000

print('Account Managers:',am)
print('Net Avg AM:',net_avg_am)
print('Region:',pd.unique(df.curr['Code Reg']))
region ref="EU"

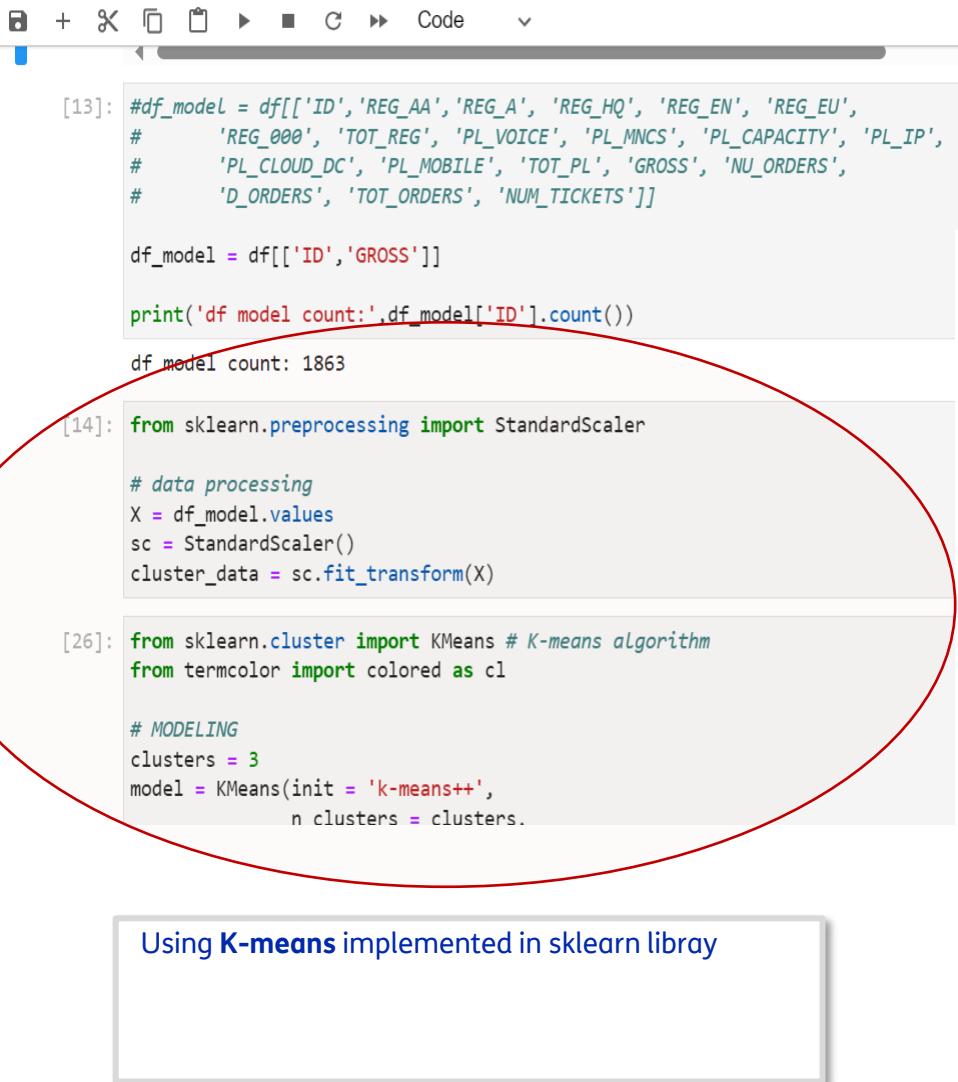
```



La piattaforma Bigdata di Sparkle e l'evoluzione IA

Palermo, Oct 2023

Big Data – Use cases implemented (ML) : Clustering



```
[13]: #df_model = df[['ID','REG_AA','REG_A', 'REG_HQ', 'REG_EN', 'REG_EU',
#                 'REG_000', 'TOT_REG', 'PL_VOICE', 'PL_MNCS', 'PL_CAPACITY', 'PL_IP',
#                 'PL_CLOUD_DC', 'PL_MOBILE', 'TOT_PL', 'GROSS', 'NU_ORDERS',
#                 'D_ORDERS', 'TOT_ORDERS', 'NUM_TICKETS']]

df_model = df[['ID','GROSS']]

print('df model count:',df_model['ID'].count())

df model count: 1863

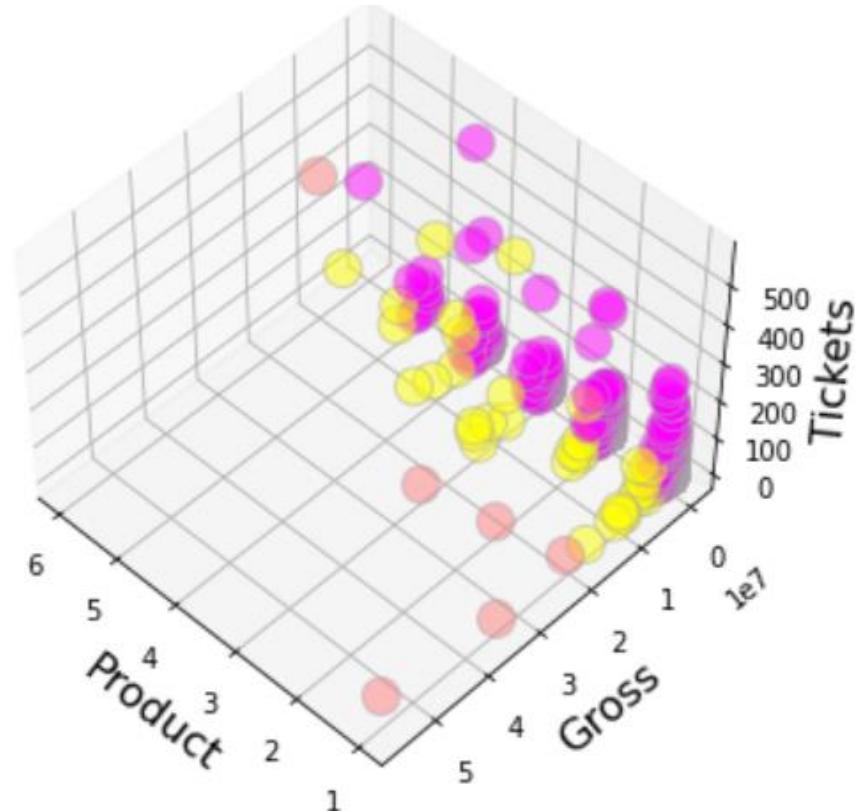
[14]: from sklearn.preprocessing import StandardScaler

# data processing
X = df_model.values
sc = StandardScaler()
cluster_data = sc.fit_transform(X)

[26]: from sklearn.cluster import KMeans # K-means algorithm
from termcolor import colored as cl

# MODELING
clusters = 3
model = KMeans(init = 'k-means++',
                n_clusters = clusters,
```

Using **K-means** implemented in **sklearn** library



Big Data – Use cases implemented (ML) : Customer Churn

Python 3 (ipykernel)

```
[1]: ID CUSTOMER_ID REG_AA REG_A REG_HQ REG_EN REG_EU REG_000 TOT_REG PL_VOICE ... PL_IP PL_CLOUD_DC PL_MOBILE
0 1 BNCPBM 0 0 0 1 0 0 1 0 ... 0 0 0 0
1 2 JTSTRM 0 0 0 1 0 0 1 0 ... 0 0 0 0
2 3 RVSNSR 0 0 0 1 0 0 1 0 ... 0 0 0 0
3 4 XPRNTR 0 0 0 1 0 0 1 0 ... 1 0 0 0
4 5 KPCLDH 1 0 0 0 0 0 1 0 ... 1 0 0 0
5 rows × 22 columns
```

```
[2]: # init setup
from pycaret.classification import *
s = setup(data, target = 'CHURN', ignore_features = ['CUSTOMER_ID'])
```

Comparing model performances

```
[3]: # compare all models
best_model = compare_models(sort='AUC')
```

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	0.9502	0.8768	0.2233	0.3417	0.2625	0.2394	0.2483	0.0680
lightgbm	0.9462	0.8714	0.2100	0.2933	0.2402	0.2146	0.2195	0.1400
xgboost	0.9533	0.8688	0.2600	0.3950	0.3122	0.2900	0.2972	0.6070
ada	0.9581	0.8594	0.1933	0.4333	0.2610	0.2479	0.2733	0.0600
rf	0.9462	0.8388	0.2233	0.3193	0.2569	0.2306	0.2372	0.1100
lda	0.9415	0.7823	0.1533	0.2019	0.1697	0.1435	0.1468	0.0140
lr	0.9573	0.7603	0.0000	0.0000	0.0000	-0.0013	-0.0018	1.9020
knn	0.9439	0.7238	0.0000	0.0000	0.0000	-0.0184	-0.0202	0.0170
dt	0.9312	0.6374	0.2967	0.2442	0.2635	0.2297	0.2323	0.0090
et	0.9359	0.6340	0.2233	0.3053	0.2316	0.2000	0.2142	0.0900
nb	0.9117	0.5418	0.0833	0.0072	0.0133	0.0051	0.0130	0.0070
qda	0.9581	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0130
svm	0.9565	0.0000	0.0600	0.3000	0.1000	0.0912	0.1248	0.0080
ridge	0.9573	0.0000	0.0000	0.0000	0.0000	-0.0013	-0.0018	0.0070

Pycaret is an open-source low code ML library in Python that automates ML workflow.

Big Data – Improve productivity using GenAI {1/2}

```
val conf=new SparkConf().setAppName(appName);
val sc=new SparkContext(conf);
val spark = SparkSession.builder.config(conf).getOrCreate();

val schema = new StructType().
  add("timestamp_cdr", "long").
  add("type", "integer").
  add("observation-domain", "integer").
  add("observation-point-id", "long").
  ...

val df = spark
  .readStream
  .option("sep", "|")
  .schema(schema)
  .csv(inputPath)
  .withColumn("file_name", input_file_name())

val dfj = df.join(dfDecodeOp, df("tcap-local-operation-code") === dfDecodeOp("code"), "leftouter")
```

- ✓ Spark Job written in Scala;
- ✓ Two output stream that writes on hdfs count aggregations of csv input stream.

```
val queryDF = dfj.writeStream
  .foreachBatch { (batchDF: DataFrame, batchId: Long) =>
    batchDF.persist()
    batchDF.groupBy(
      date_format(col("timestamp_cdr").cast("bigint") / 1000).cast("timestamp"), "yyyy/MM/dd HH").alias("Time"),
      col("mtcp3-opc"),
      col("mtcp3-dpc"),
      ...
      when(col("tcap-local-operation-code").isNull, value = "No data").when(col("operation").isNull, value = "No
      data").otherwise(col("operation")).as("tcap-local-operation-code"),
      col("tcap-message-type")
    ).count().withColumnRenamed("count", "ALL").coalesce(1).
    write.mode("append").
    option("compression", "gzip").
    option("header", true).
    csv(outputPath)

    batchDF.unpersist()
  }
  .trigger(Trigger.ProcessingTime(triggerProcessingTime))
  .option("checkpointLocation", checkPointPath)
  .start()
```

```
val queryDFStat = dfj.writeStream
  .foreachBatch { (batchDF: DataFrame, batchId: Long) =>
    batchDF.persist()
    batchDF.groupBy(
      date_format(current_timestamp(), "yyyy-MM-dd").as("date_msu"),
      date_format(current_timestamp(), "yyyy-MM-dd HH:mm:ss").as("timestamp"),
      functions.reverse(functions.split(col("file_name"), "/"))(0).as("file_name"),
      ...
      when(col("tcap-local-operation-code").isNull, value = "No data").when(col("operation").isNull,
      value = "No data").
      otherwise(col("operation")).as("tcap-local-operation")
    ).count().coalesce(1).
    write.mode("append").
    option("header", true).
    csv(statPath)
    batchDF.unpersist()
  }
  .trigger(Trigger.ProcessingTime(triggerProcessingTime))
  .option("checkpointLocation", statCheckPointPath)
  .start()
```

Big Data – Improve productivity using GenAI {2/2}



scrivi un job spark in scala che definisce uno stream su un folder contenente file csv

```
val spark = SparkSession  
  .builder()  
  .appName("CSVFileStreamExample")  
  .getOrCreate()  
  
// Definisci lo schema dei dati CSV  
val schema = StructType(  
  List(  
    StructField("col1", StringType, true),  
    StructField("col2", IntegerType, true)  
  )  
)  
  
....  
  
// Crea uno stream su questa directory  
val stream = spark.readStream  
  .schema(schema)  
  .option("header", "true") // Se la prima riga contiene l'intestazione  
  .csv(csvDirectory)
```



il separatore all'interno dei csv è il |

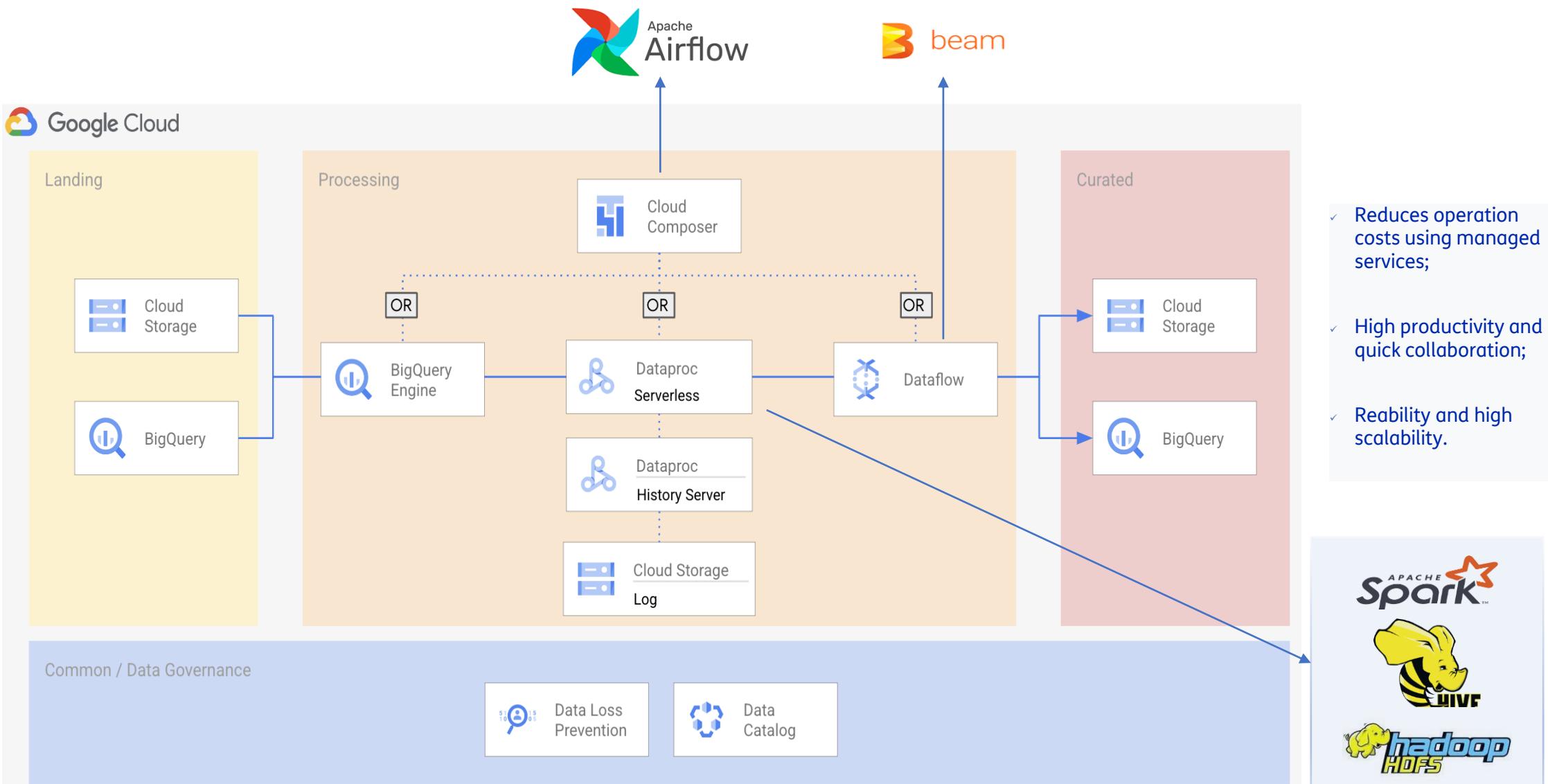
```
....  
// Crea uno stream su questa directory, specificando il separatore "|"  
val stream = spark.readStream  
  .schema(schema)  
  .option("header", "true") // Se la prima riga contiene l'intestazione  
  .option("delimiter", "|") // Specifica il separatore  
  .csv(csvDirectory)
```



scrivi due stream in output che effettuano la count rispettivamente su col1 e col2 e scrivono in due differenti folder

```
// Esegui la count su "col1" e scrivi i risultati in una directory  
val countCol1Stream = stream  
  .groupBy("col1")  
  .count()  
  .writeStream  
  .outputMode("complete") // Usa "complete" in modo da aggiornare la count completa a ogni micro-batch  
  .format("console") // Puoi modificare il formato di output a seconda delle esigenze  
  .option("checkpointLocation", "path/to/checkpoint1")  
  .start("path/to/output1")  
  
// Esegui la count su "col2" e scrivi i risultati in un'altra directory  
val countCol2Stream = stream  
  .groupBy("col2")  
  .count()  
  .writeStream  
  .outputMode("complete") // Usa "complete" in modo da aggiornare la count completa a ogni micro-batch  
  .format("console") // Puoi modificare il formato di output a seconda delle esigenze  
  .option("checkpointLocation", "path/to/checkpoint2")  
  .start("path/to/output2")  
  
// Attendi fino a quando entrambi i processi di streaming non vengono terminati  
countCol1Stream.awaitTermination()  
countCol2Stream.awaitTermination()
```

Big Data – Data Platform @ GCP



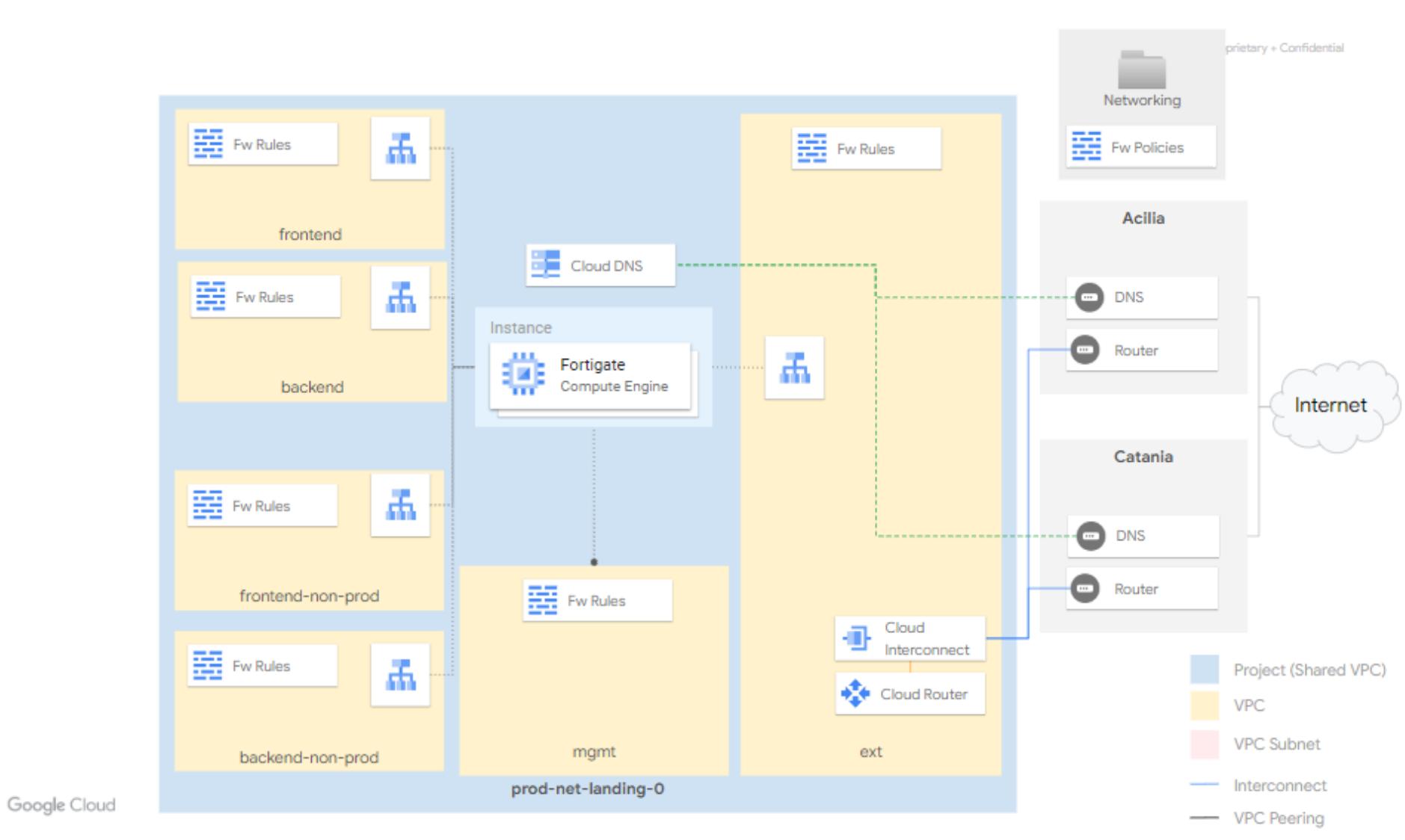
Big Data – Data Platform @ GCP – Infrastructure as code

The screenshot shows a Visual Studio Code interface with several tabs open. The active tab is `vpc-backend-nonprod.tf`, which contains Terraform code for creating a VPC. The code defines two subnets: `backend-nonprod-default` and `backend-nonprod-dataproc`. The `backend-nonprod-default` subnet is located in the `europe-west8` region with an IP range of `172.28.64.0/24`. The `backend-nonprod-dataproc` subnet is also in the `europe-west8` region with an IP range of `172.28.66.0/24` and includes secondary IP ranges for `backend-nonprod-pod` and `backend-nonprod-services`.

```
17 # backend-nonprod VPC
18
19 module "backend-nonprod-vpc" {
20   source      = "../modules/net-vpc"
21   project_id = module.net-project.project_id
22   name        = "backend-nonprod"
23   mtu         = 1500
24   dns_policy = {
25     inbound = false
26     logging = false
27   }
28   create_googleapis_routes = null
29   subnets = [
30     {
31       ip_cidr_range = "172.28.64.0/24"
32       name          = "backend-nonprod-default"
33       region        = "europe-west8"
34     },
35     {
36       name          = "backend-nonprod-dataproc"
37       region        = "europe-west8"
38       ip_cidr_range = "172.28.66.0/24"
39       secondary_ip_ranges = {
40         backend-nonprod-pod = "100.64.0.0/16"
41         backend-nonprod-services = "172.28.67.0/24"
42       }
43     },
44   ]
45 }
```

- ✓ allows teams to define and manage infrastructure using version control, which makes it easier for multiple people to collaborate and work on the same codebase;
- ✓ allows to define infrastructure using a high-level configuration language.

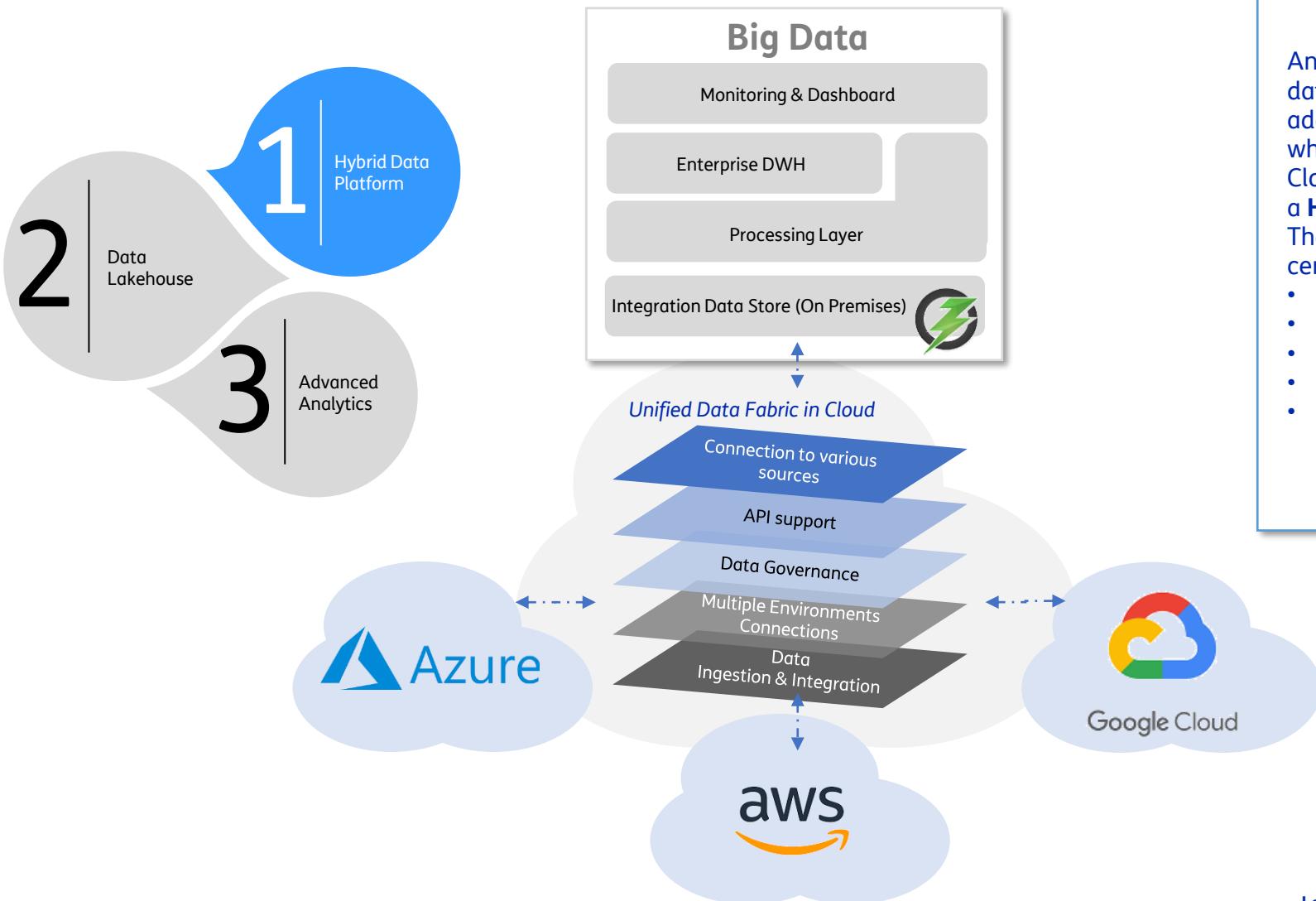
Big Data – Network and security on GCP



- ✓ 6 security context (VPC):
- ✓ 2 x frontend (prod, nonprod);
- ✓ 2 x backend (prod,nonprod);
- ✓ 1 management;
- ✓ 1 external (on-promise connectivity)
- ✓ Fortigate group to manage FW rules between external context and internal contexts

Next Step – Big Data Transformation

Advantages of the **Hybrid cloud**: scalability, security and flexibility



An important step in becoming a data-driven organization will be the adoption of a **Unified Data Fabric**, which will transform the Private Cloud architecture of Big Data into a **Hybrid Cloud architecture**.

The following functions will be centralized and simplified:

- Data Catalog
- Data governance
- Data security
- Centralized user provisioning
- Centralized environments provisioning

Next Step – Lakehouse Trend

"Data Lakehouses integrate and unify on a single platform the capabilities of data warehouses and data lakes, aiming to support AI, BI, ML and data engineering"

Gartner Research January 2022



Benefits Data Lakehouse



A single data repository requires less time and budget to administer

Direct access to broader dataset for BI tools and ML analysis

Advanced Analytics like Real time & Streaming Analytics

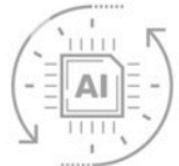
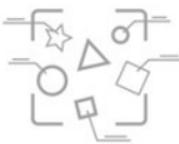
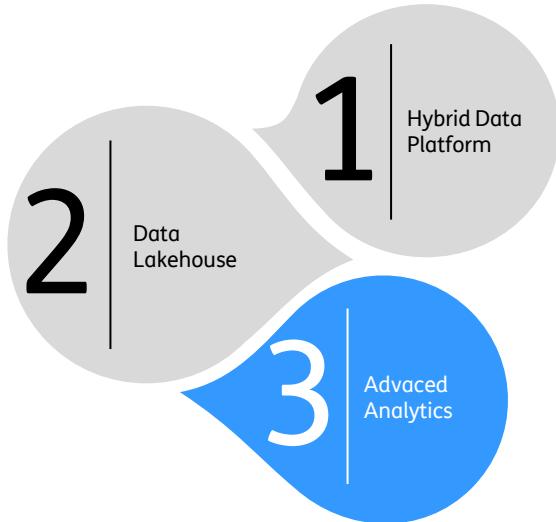
Simplified data governance due to having a single control point

Less data redundancy

ACID-compliant transaction support.

Next Step – Advanced Analytics

Mission: give voice to data!!



Correlation Analysis and Pattern identification

- Clustering and Correlation ML algorithms to identify **event correlation and patterns**
- Detect **hidden phenomena**, such as abandonment proneness

Insights & predictions

- **Ensemble of advanced ML models** for defining business insights
- Data Classification e Data Regression
- GeoAnalysis



Business Value

Support on strategic business choices to different company and market players



Speed & Quality

Ready to go! Increased time to market with high quality and efficiency



AI&ML Capabilities

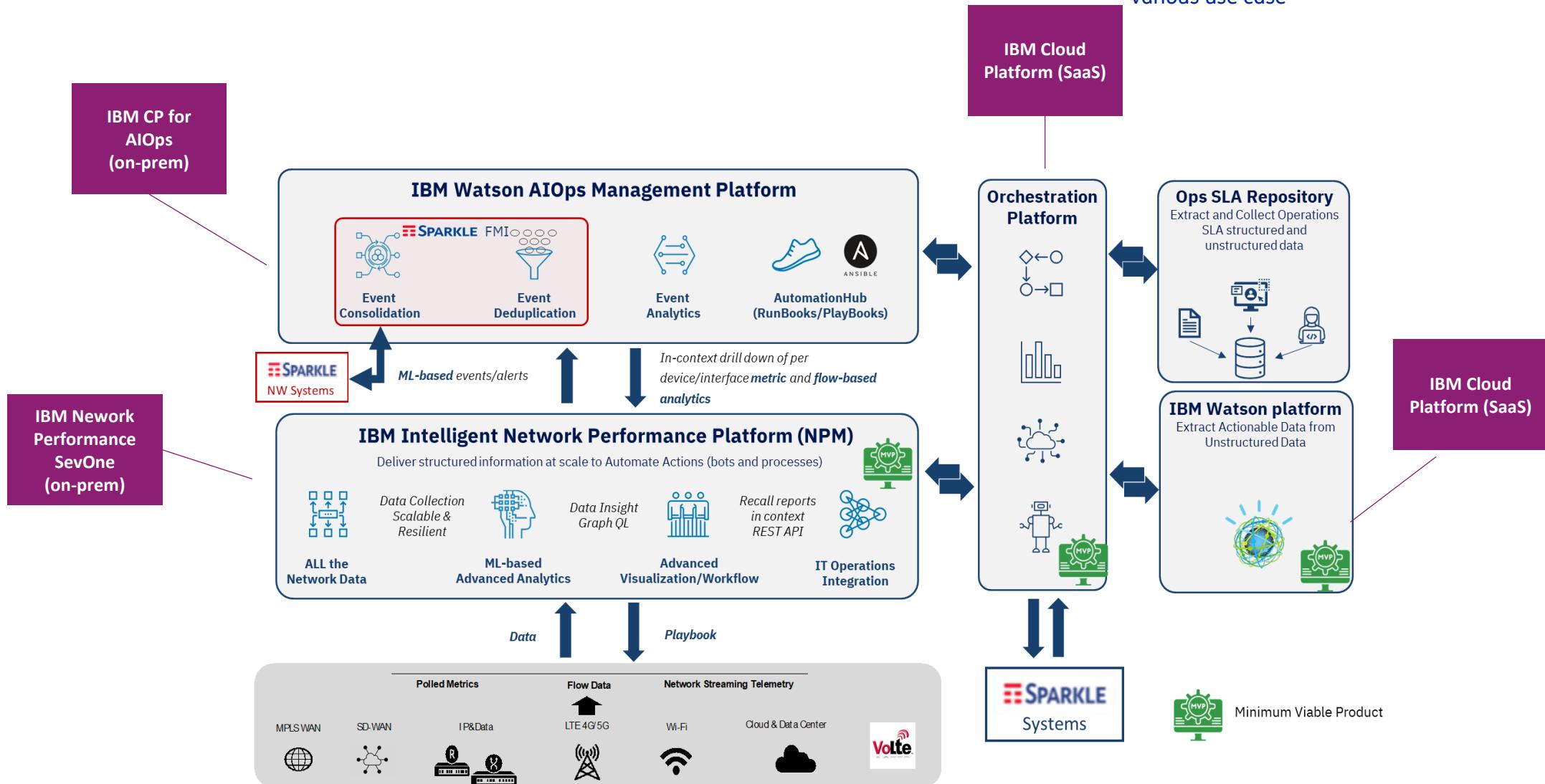
Hidden phenomena and unknown cases identification

AI Networks : Use Cases

Service Problem Management			
Process autom. tools	Incident Automation – smart view e-mail Mgmt	Proactive Customer Notification	RFO Management
SLA Mng.	Fault & Perf. Mng	Closed-loop automation	Network AIOps
Repetitive tasks automation Automation of repetitive tasks on the BB IP network (software updates, tests, massive configurations, security patches)	Incident Automation – smart view e-mail Mgmt Improve the management of customer/supplier communications through automation capabilities.	Proactive Customer Notification Due to a Major Fault, it is required to proactively inform the customer of the failure of the internal network	RFO Management Production of a Report that contains ticket data (e.g. automatic management of RFO Customer requests)
SLA Management Visibility of SLAs across all systems. For example, visibility of all individual services, for provisioning and assurance	IP Network-Monitoring IP network monitoring for network degradation/ saturation and consequent automatic notification to the Customers involved	Closed-loop automation Automatic Remediation SPANLOSS Mgmt: Automatic adjustment of the transmission laser power due to high signal attenuation Automatic Rerouting of transmission circuits in case of Major Fault	Fault prevention Fault prediction, capture and processing of weak signals on the network in order to prevent faults

AI Networks : Architecture

- ✓ Solution is composed of specialized elements that interact each other in a flexible way in order to follow an agile approach in the implementation/direction of the various use case



Q&A



GRAZIE