

Architect

SACC

2022 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2022

· 激发架构性能 点亮业务活力

云上会议 网络直播 | 2022年10月27-29日

IT168.com

ChinaUnix.net

ITPUB

# Byzer - 一门面向 Data+AI 的云原生低 代码化语言设计哲学

祝海林 | Byzer PMC & Kyligence 技术合伙人  
hailin.zhu@kyligence.io



祝海林

[hailin.zhu@kyligence.io](mailto:hailin.zhu@kyligence.io)

- 2008年第 1 次创业
- 2012 年发布了自己的第 1 个重要开源项目「ServiceFramework」
- 2016年发布自己第 2 个重要开源项目「Byzer」
- 目前在维护项目多达几十个

2021 - 至今	资深数据工程师	Byzer PMC, Kyligence 技术合伙人
2017 - 2020	资深数据工程师	丁香园
2015 - 2016	数据工程师	乐视云



# KYLIGENCE 公司介绍

自主开源技术，打造开源生态



- 全球领先的大数据 OLAP 领导者
- 中国首个 Apache 顶级开源项目
- 1500+ 全球生产用户



- 面向 Data + AI 的类 SQL 语言
- 云原生，覆盖桌面端到服务端
- 金融、互联网等行业应用案例



# 了解 Byzer



Why

大数据 & AI  
平台落地困难 成本高



How

从语言层面解决  
问题



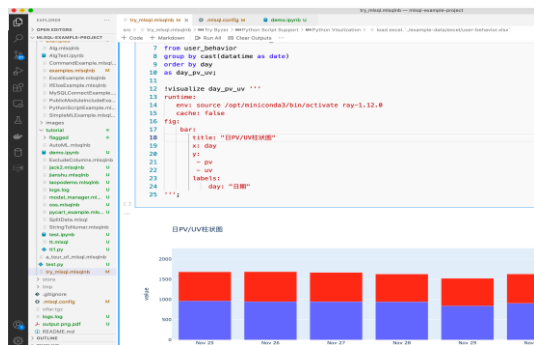
What

面向 Data+AI 领域  
云原生类 SQL 语言

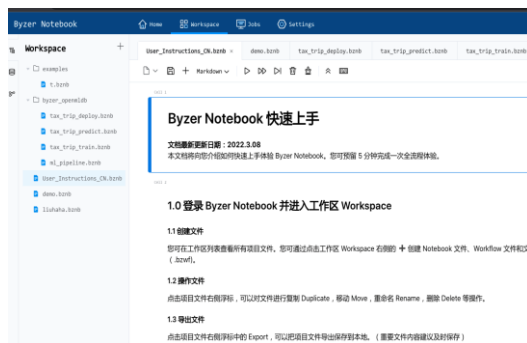
# Byzer 交互形态

桌面版 开箱即用

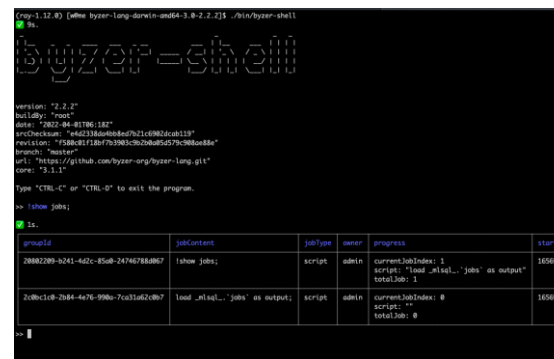
支持  
Windows/MAC/Linux



Web IDE

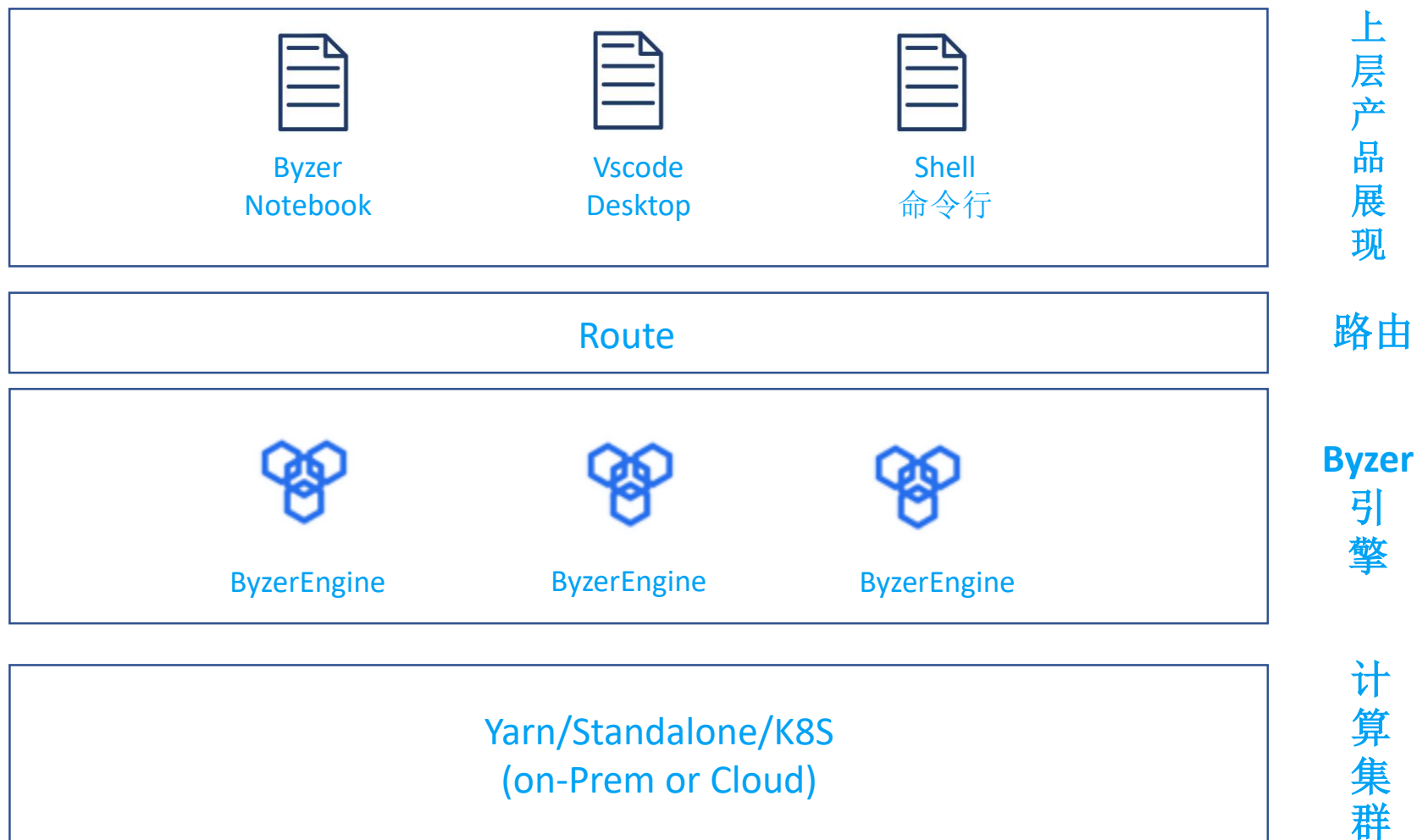


Byzer-shell /Byzer CLI

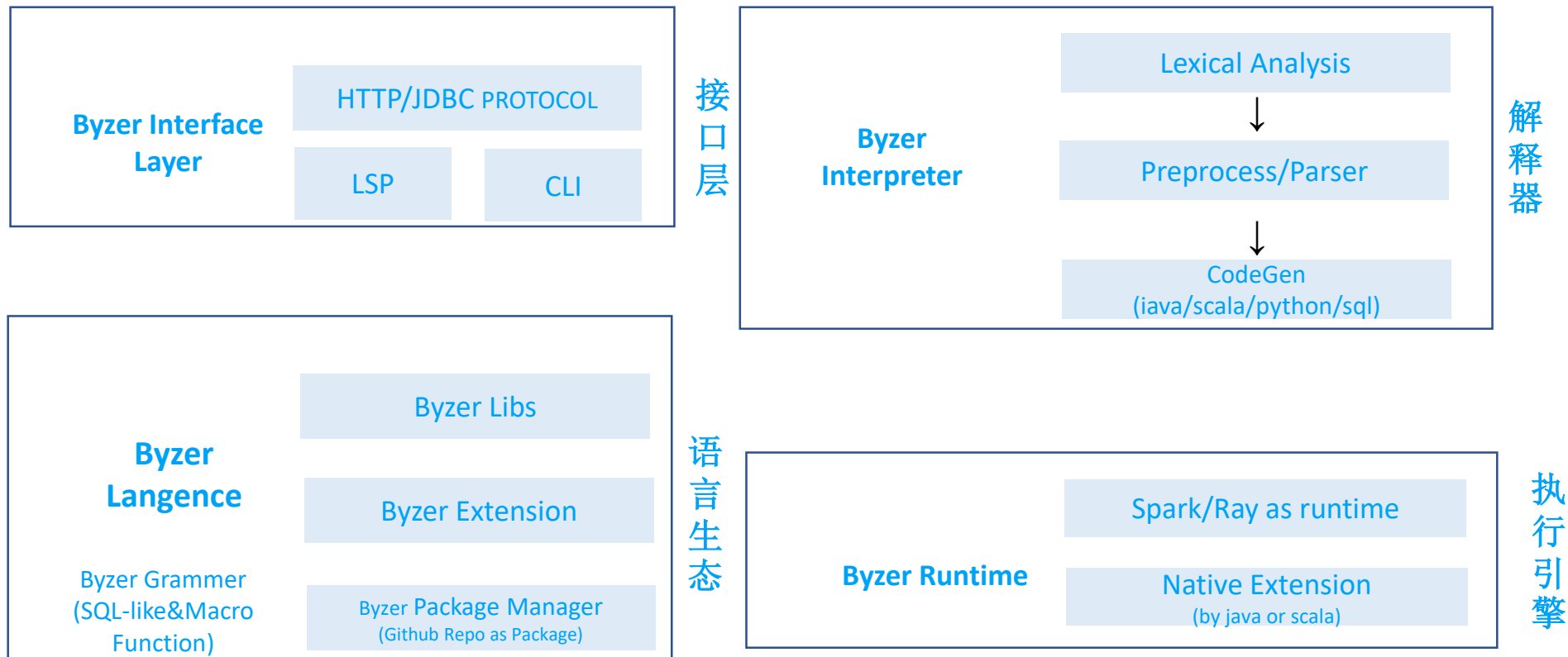


# Byzer 架构图

## Byzer Architecture 产品层

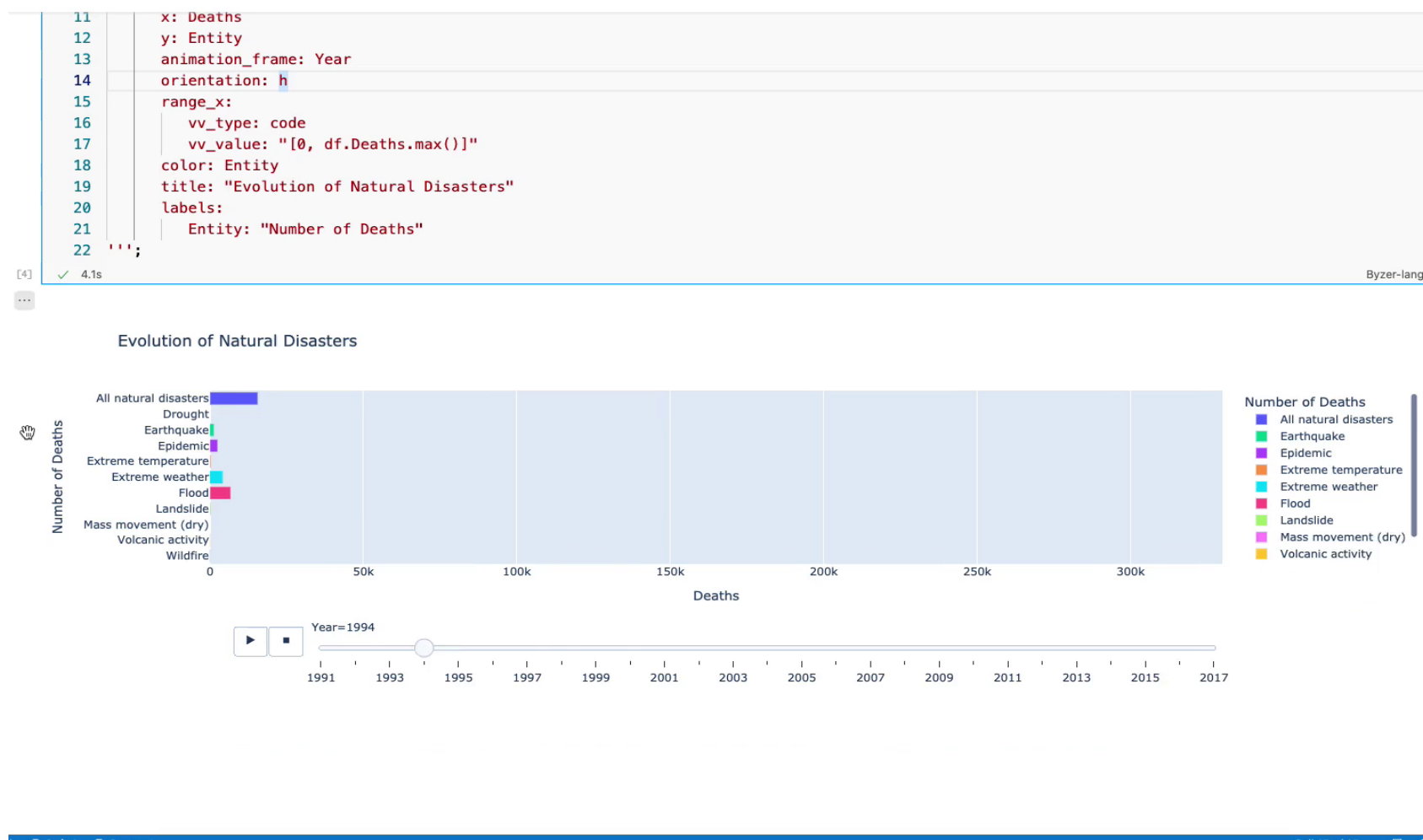


## Byzer Langence Architecture 引擎层





# Byzer 自然灾害可视化演示视频



# 示例回顾：Byzer 语言设计特点

将数据抽取为表格形式

表格化

使用 SQL 进行数据处理

SQL 化

使用 Yaml 文本绘图

可视化

特征工程/模型转化为函数

函数化

# Byzer 机器学习示例

The screenshot shows the VS Code editor with the following components:

- EXPLORER:** Shows the project structure with folders like `.vscode`, `data`, `logs`, `spark-warehouse`, and `src`. The `src` folder is expanded, showing files like `mock_data.mlsq`, `AutoML.mlsqInb`, `example.mlsqInb`, `SimpleMLExa...`, `test.ipynb`, and `test.mlsqInb`.
- Editor:** Displays the content of `SimpleMLExample.mlsqInb`, which is a Spark SQL query. The query is as follows:

```
1 train trainData as AutoML.`/tmp/auto_ml_model` where
2 algos="GBTs,LinearRegression,LogisticRegression,NaiveBayes,RandomFo
3 and keepVersion="true"
4 and evaluateTable="testData";
```
- Table:** A table showing the configuration for the query. It has two columns: `name` and `value`.

name	value
modelPath	__auto_ml_RandomForest/_model_1/model/0

# 示例回顾：Byzer 机器学习特点

覆盖整个机器学习流程

全流程

支持类SQL语法，简单易学

类 SQL

数据抽取，处理，特征工程，模型训练全程分布式

真分布式

覆盖深度学习（on Byzer-python）

深度学习

# Byzer-python 示例

EXPLORER

OPEN E...  
SimpleML... M  
AutoML.mlsqInb...  
example.mlsqIn...  
.mlsql.config M

BYZER-PYTHON  
> .vscode  
> data  
> logs  
> spark-warehouse  
src  
common  
mock\_data.mlsq  
AutoML.mlsqInb  
example.mlsqInb  
SimpleMLExa... M  
test.ipynb  
test.mlsqInb  
store  
.gitignore  
mlsql.config M

SimpleMLExample.mlsqInb M AutoML.mlsqInb example.mlsqInb X .mlsql.config M

src > example.mlsqInb > M使用Python做分布式数据处理 > #python  
+ Code + Markdown | ▶ Run All ≡ Clear Outputs ...

自动将SQL 表转化为pandas, 进而进行操作。这种适合数据已经被聚合, 数据规模不大

```
1 #python
2 #input=iris
3 #dataMode=model
4 #schema=st(field(content,string),field(mime,string))
5 #env=source /opt/miniconda3/bin/activate ray-1.12.0
6
7 from pyjava.api.mlsq import RayContext,PythonContext
8 import plotly.express as px
9 from sklearn.decomposition import PCA
10
```



# 示例回顾: Byzer-python特点

SQL 和 Python 代码共存

Not ebook 化

SQL 和 Python 数据互访

数据互通

SQL 和 Python 全分布式执行

真分布式

分布式DataFrame: Dask

分布式  
Data Frame

# Byzer-python 示例

EXPLORER

OPEN E...  
SimpleML... M  
AutoML.mlsqInb...  
example.mlsqInb...  
.mlsql.config M

BYZER-PYTHON  
.vscode  
data  
logs  
spark-warehouse  
src  
common  
mock\_data.mlsqInb  
AutoML.mlsqInb  
example.mlsqInb  
SimpleMLExa... M  
test.ipynb  
test.mlsqInb  
store  
.gitignore  
mlsql.config M

SimpleMLExample.mlsqInb M AutoML.mlsqInb example.mlsqInb X .mlsql.config M

src > example.mlsqInb > M使用Python做分布式数据处理 > #python  
+ Code + Markdown | Run All Clear Outputs ...

自动将SQL 表转化为pandas, 进而进行操作。这种适合数据已经被聚合, 数据规模不大

```
1 #python
2 #input=iris
3 #dataMode=model
4 #schema=st(field(content,string),field(mime,string))
5 #env=source /opt/miniconda3/bin/activate ray-1.12.0
6
7 from pyjava.api.mlsqInb import RayContext,PythonContext
8 import plotly.express as px
9 from sklearn.decomposition import PCA
10
```

# 示例回顾: Byzer-python特点

SQL 和 Python 代码共存

Not e b o o k 化

SQL 和 Python 数据互访

数据互通

SQL 和 Python 全分布式执行

真分布式

分布式DataFrame: Dask

分布式  
Data Frame

# 示例回顾：Ray 带来了什么

数据处理，模型训练，全分布式

分布式

Spark/Ray 环境隔离，屏蔽Python高度复杂的环境

环境隔离

Hybrid Runtime

融合

Ray可以为作为模型部署底座

模型部署

# Byzer 的一些工程特性

模块化支持

模板支持

条件分支语句支持



Cell 2

```
1 include lib.`gitee.com/allwefantasy/lib-core`  
2 where alias="libCore";  
3  
4 include local.`libCore.udf.hello`;  
5 select hello() as name as output;
```

Result

Job Details

Log Message

name

hello world

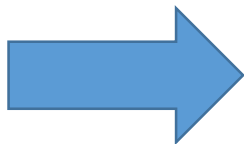
Total 1 < 1 > Go to 1



知乎 @Byzer Man

# 模板支持

```
1 select SUM( case when `id` is null or `id`='' then 1 else 0 end ) as id,
2 SUM( case when `diagnosis` is null or `diagnosis`='' then 1 else 0 end ) as diagnosis,
3 SUM( case when `radius_mean` is null or `radius_mean`='' then 1 else 0 end ) as radius_mean,
4 SUM( case when `texture_mean` is null or `texture_mean`='' then 1 else 0 end ) as texture_mean,
5 SUM( case when `perimeter_mean` is null or `perimeter_mean`='' then 1 else 0 end ) as perimeter_mean,
6 SUM( case when `area_mean` is null or `area_mean`='' then 1 else 0 end ) as area_mean,
7 SUM( case when `smoothness_mean` is null or `smoothness_mean`='' then 1 else 0 end ) as smoothness_mean,
8 SUM( case when `compactness_mean` is null or `compactness_mean`='' then 1 else 0 end ) as compactness_mean,
9 SUM( case when `concavity_mean` is null or `concavity_mean`='' then 1 else 0 end ) as concavity_mean,
10 SUM( case when `concave points_mean` is null or `concave points_mean`='' then 1 else 0 end ) as concave_pc,
11 SUM( case when `symmetry_mean` is null or `symmetry_mean`='' then 1 else 0 end ) as symmetry_mean,
12 SUM( case when `fractal_dimension_mean` is null or `fractal_dimension_mean`='' then 1 else 0 end ) as fractal_
13 SUM( case when `radius_se` is null or `radius_se`='' then 1 else 0 end ) as radius_se,
14 SUM( case when `texture_se` is null or `texture_se`='' then 1 else 0 end ) as texture_se,
15 SUM( case when `perimeter_se` is null or `perimeter_se`='' then 1 else 0 end ) as perimeter_se,
16 SUM( case when `area_se` is null or `area_se`='' then 1 else 0 end ) as area_se,
17 SUM( case when `smoothness_se` is null or `smoothness_se`='' then 1 else 0 end ) as smoothness_se,
18 SUM( case when `compactness_se` is null or `compactness_se`='' then 1 else 0 end ) as compactness_se,
19 SUM( case when `concavity_se` is null or `concavity_se`='' then 1 else 0 end ) as concavity_se,
20 SUM( case when `concave points_se` is null or `concave points_se`='' then 1 else 0 end ) as concave_points_
21 SUM( case when `symmetry_se` is null or `symmetry_se`='' then 1 else 0 end ) as symmetry_se,
22 SUM( case when `fractal_dimension_se` is null or `fractal_dimension_se`='' then 1 else 0 end ) as fractal_
23 SUM( case when `radius_worst` is null or `radius_worst`='' then 1 else 0 end ) as radius_worst,
24 SUM( case when `texture_worst` is null or `texture_worst`='' then 1 else 0 end ) as texture_worst,
25 SUM( case when `perimeter_worst` is null or `perimeter_worst`='' then 1 else 0 end ) as perimeter_worst,
26 SUM( case when `area_worst` is null or `area_worst`='' then 1 else 0 end ) as area_worst,
27 SUM( case when `smoothness_worst` is null or `smoothness_worst`='' then 1 else 0 end ) as smoothness_worst,
28 SUM( case when `compactness_worst` is null or `compactness_worst`='' then 1 else 0 end ) as compactness_wc,
29 SUM( case when `concavity_worst` is null or `concavity_worst`='' then 1 else 0 end ) as concavity_worst,
30 SUM( case when `concave points_worst` is null or `concave points_worst`='' then 1 else 0 end ) as concave_
31 SUM( case when `symmetry_worst` is null or `symmetry_worst`='' then 1 else 0 end ) as symmetry_worst.
```



```
4 -- select concat_ws(",",collect_list(col_name)) from table3 as output;
5 select
6 #set($list=$newColumns.split(","))
7 #set($max = $list.size() - 1)
8 #foreach($i in [ $max .. 0 ])
9     #if($i == 0)
10         $list[$i]
11     #else
12         $list[$i],
13     #end
14 #end
15 from table2 as output;
16
17
```

[14] ✓ 0.1s

name	age
elena	1
candy	1
bob	1

```
select 1 as a as mockTable;  
set b_count=`select count(*) from mockTable ` where type="sql"  
  
!if '' :b_count > 1 '' ;  
    select 1 as a as final_table;  
!else;  
    select 2 as a as final_table;  
!fi;  
  
select * from final_table as output;
```

# Byzer 案例介绍



# 案例 1 – 某消费金融公司落地数据平台案例

使用 Byzer 完成大数据中心平台的迁移和升级

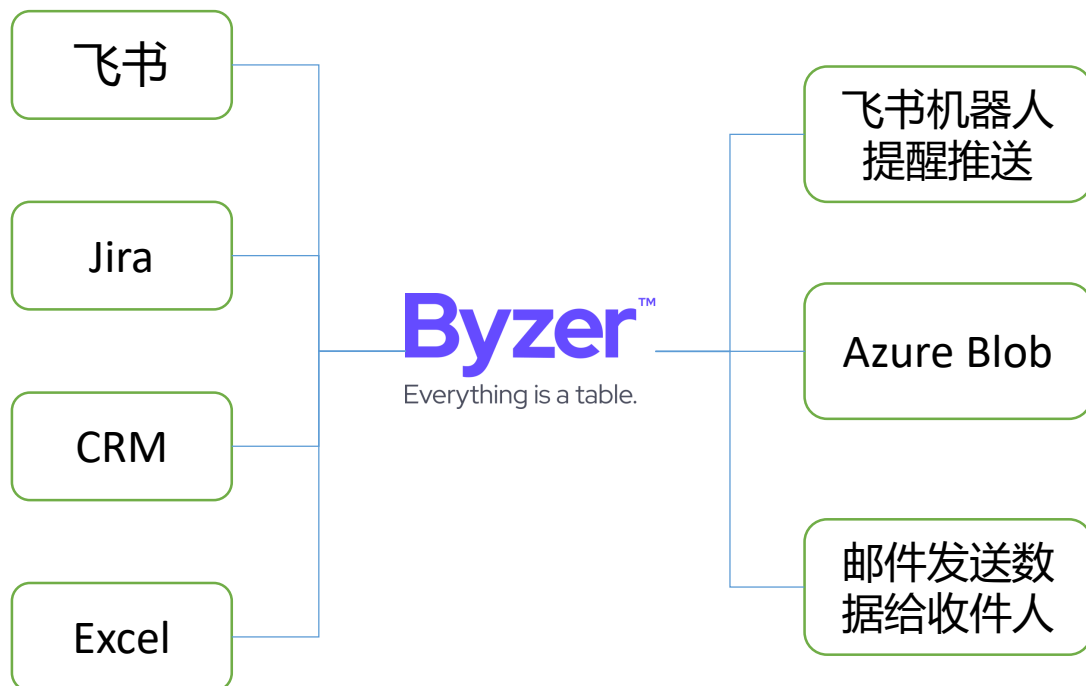
- 使用 Byzer 在语言层面统一技术栈和架构
- 生产环境稳定运行 **3 年多**
- 累计执行的数据处理任务 **700 万次**
- 单日执行的调度任务超过 **4000 个**
- 平台服务用户日活 **50+**
- **只投入了2人的研发团队**



+ Byzer™  
Everything is a table.

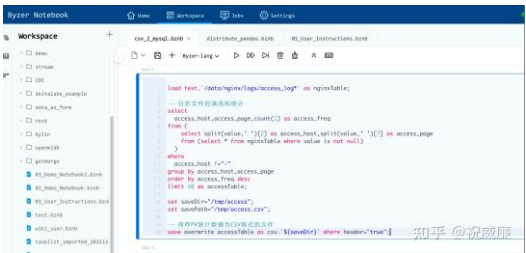
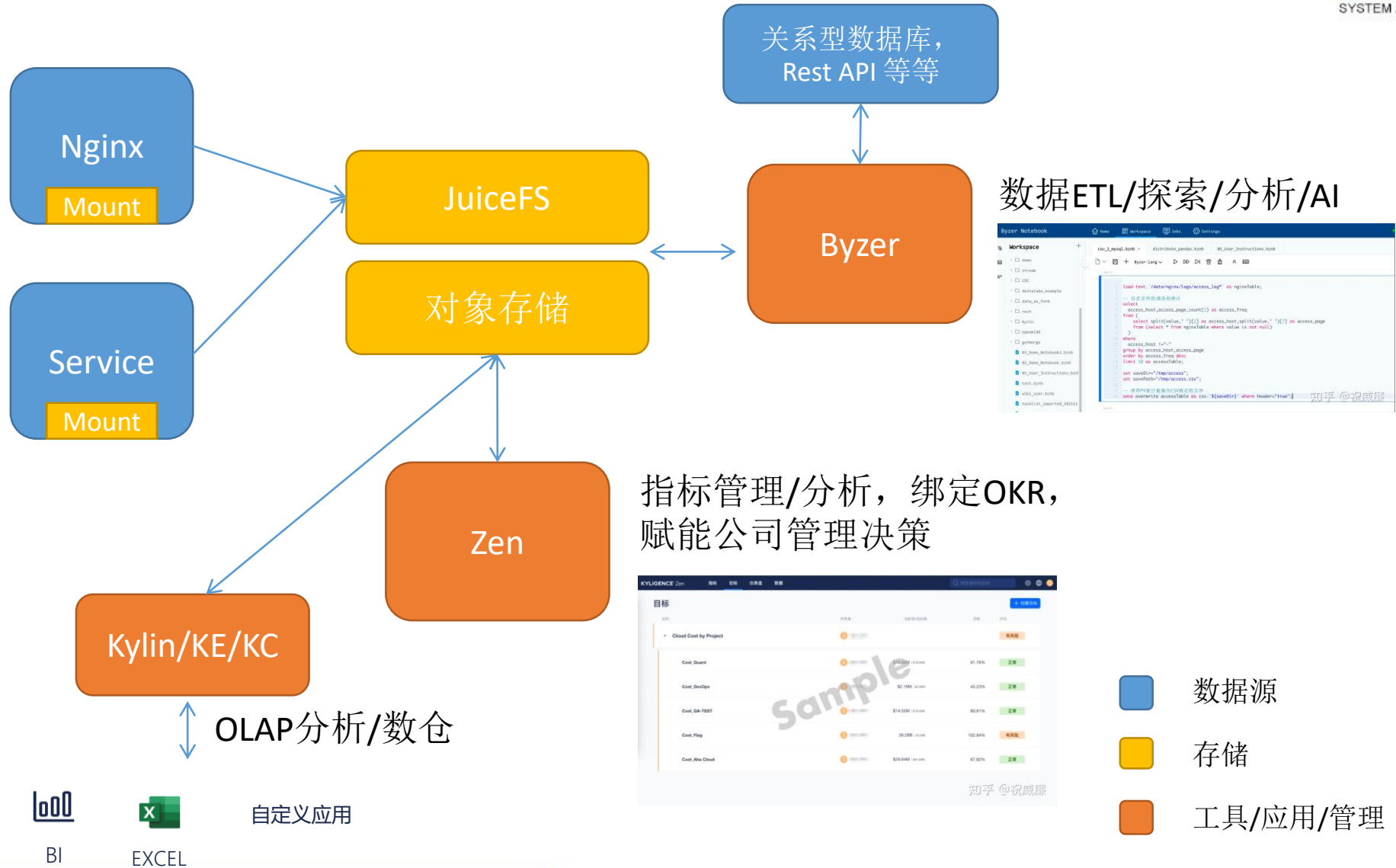


## 案例 2 - Kyligence 工时管理分析系统



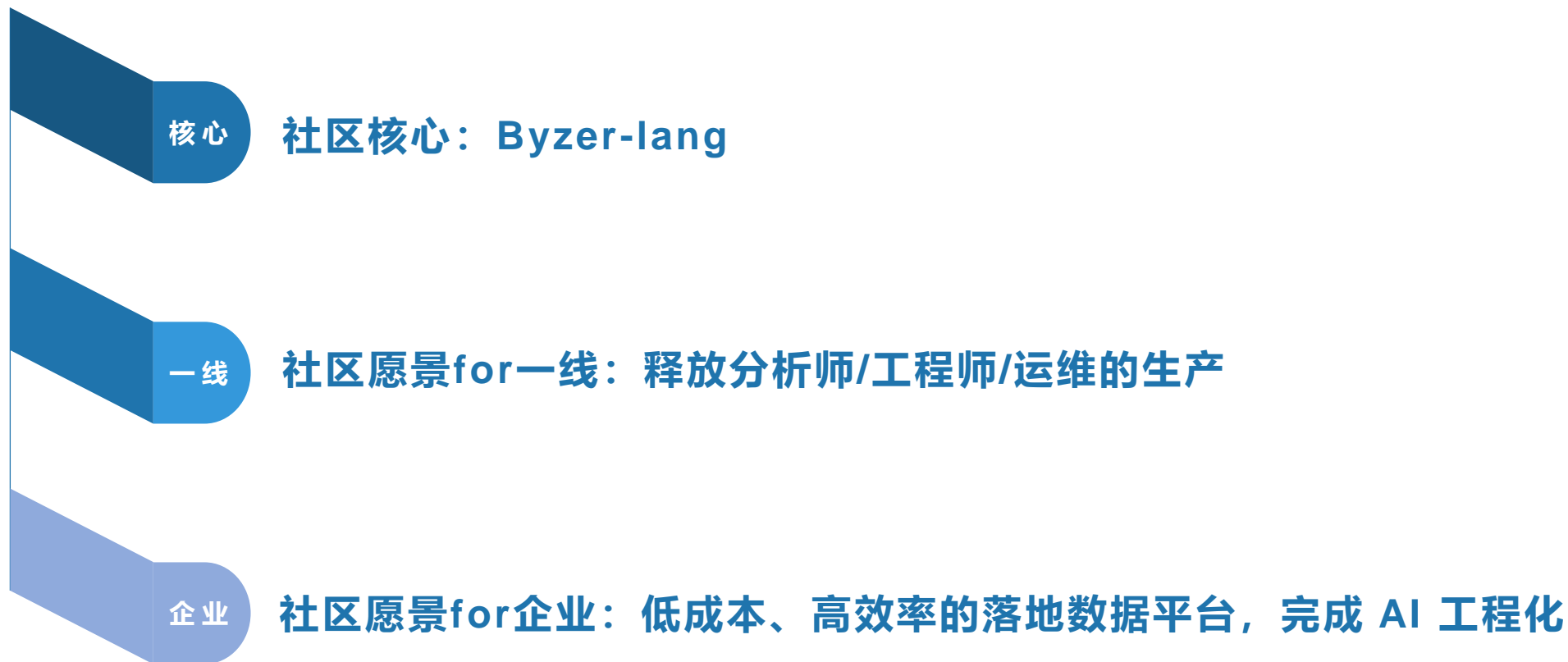
- 通过 Byzer 获取不同 SaaS 系统的数据，进行融合分析
- 结果会根据不同的用途发送给不同的系统
- 生产环境每天定时更新，数据定时推送
- 管理者需要评估不同项目的 ROI
- 各部门的工时信息记录在不同的业务系统，但项目在不同的系统之间有交叉

案例 3 – 如何用最简单的方式构建大数据和AI平台



# Byzer 开源社区：面向 Data&AI 打造技术生态

<https://github.com/byzer-org>





THANKS

Armed