

Masterclass Smart Data Fabric in Practice: Interoperabilität mit Embedded Python

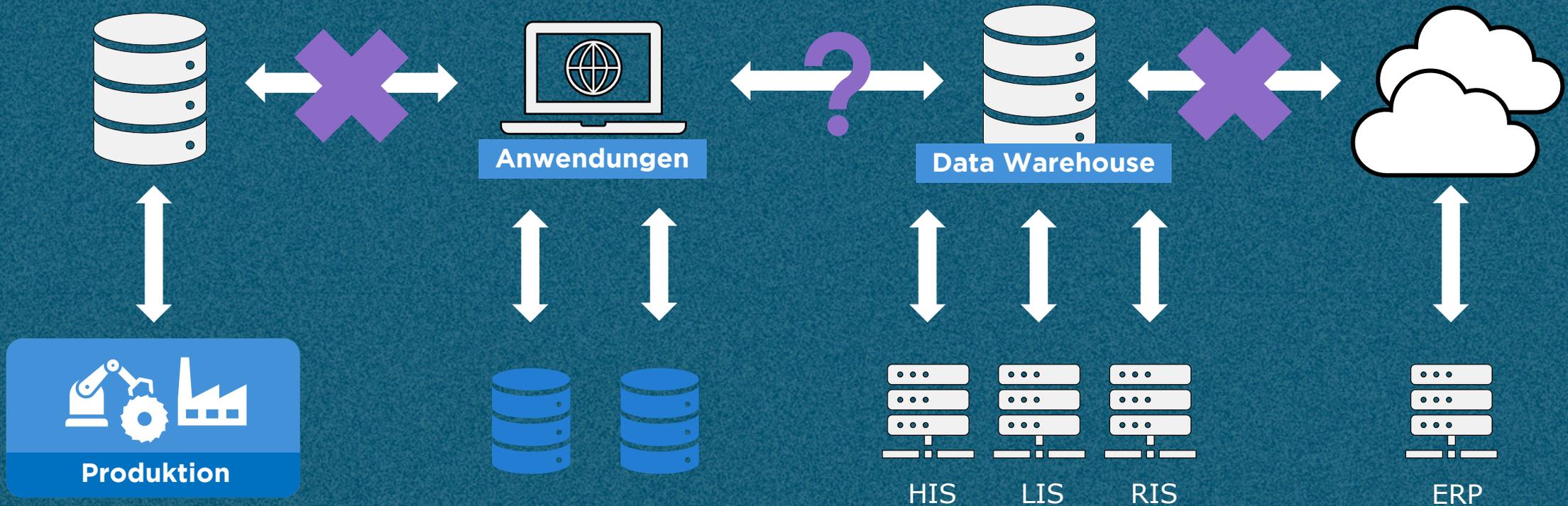
Jannis Stegmann | InterSystems
Felix Vetter | InterSystems



Status Quo in der Datenlandschaft vieler Unternehmen?

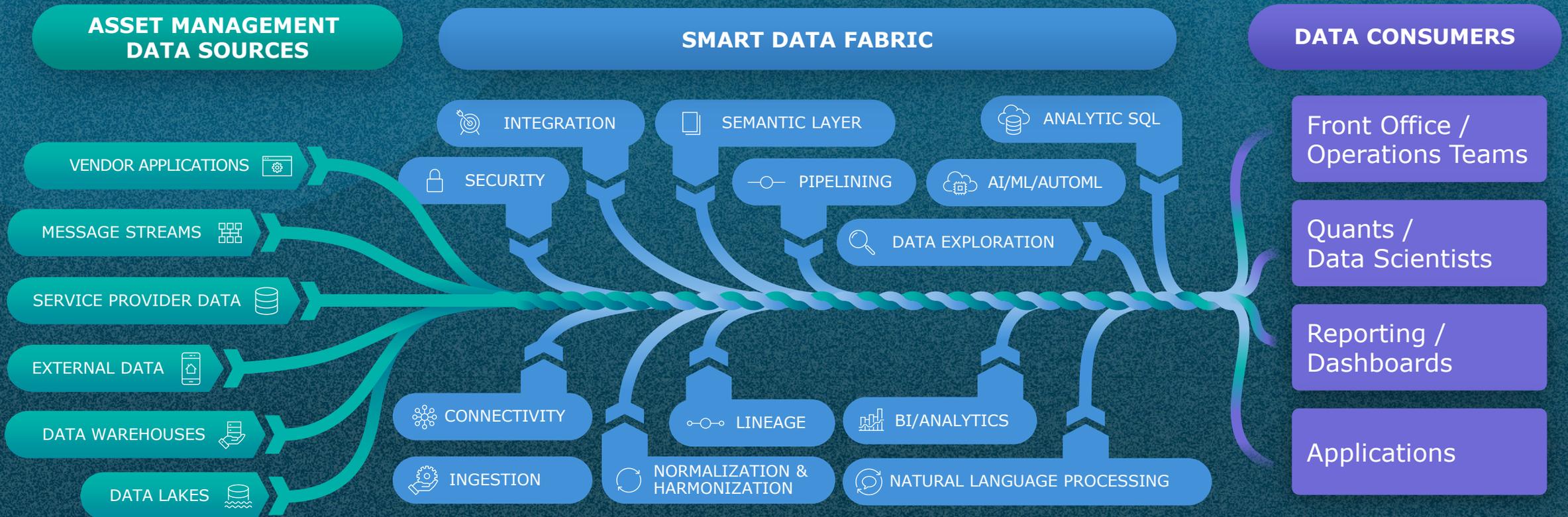


Wie können Unternehmensentscheidungen verbessert und beschleunigt werden?



Smart Data Fabric

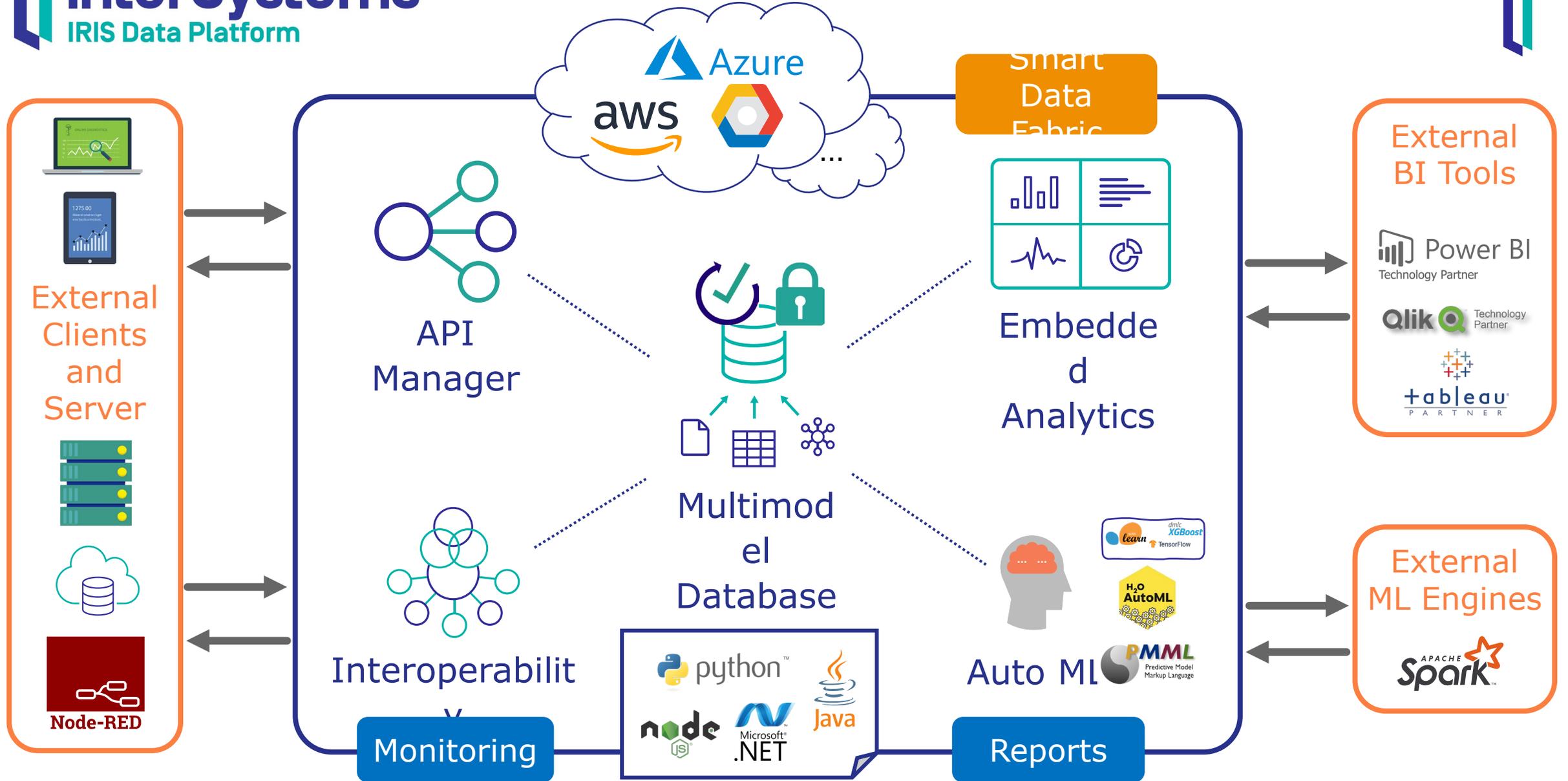
Powered by InterSystems IRIS Data Platform



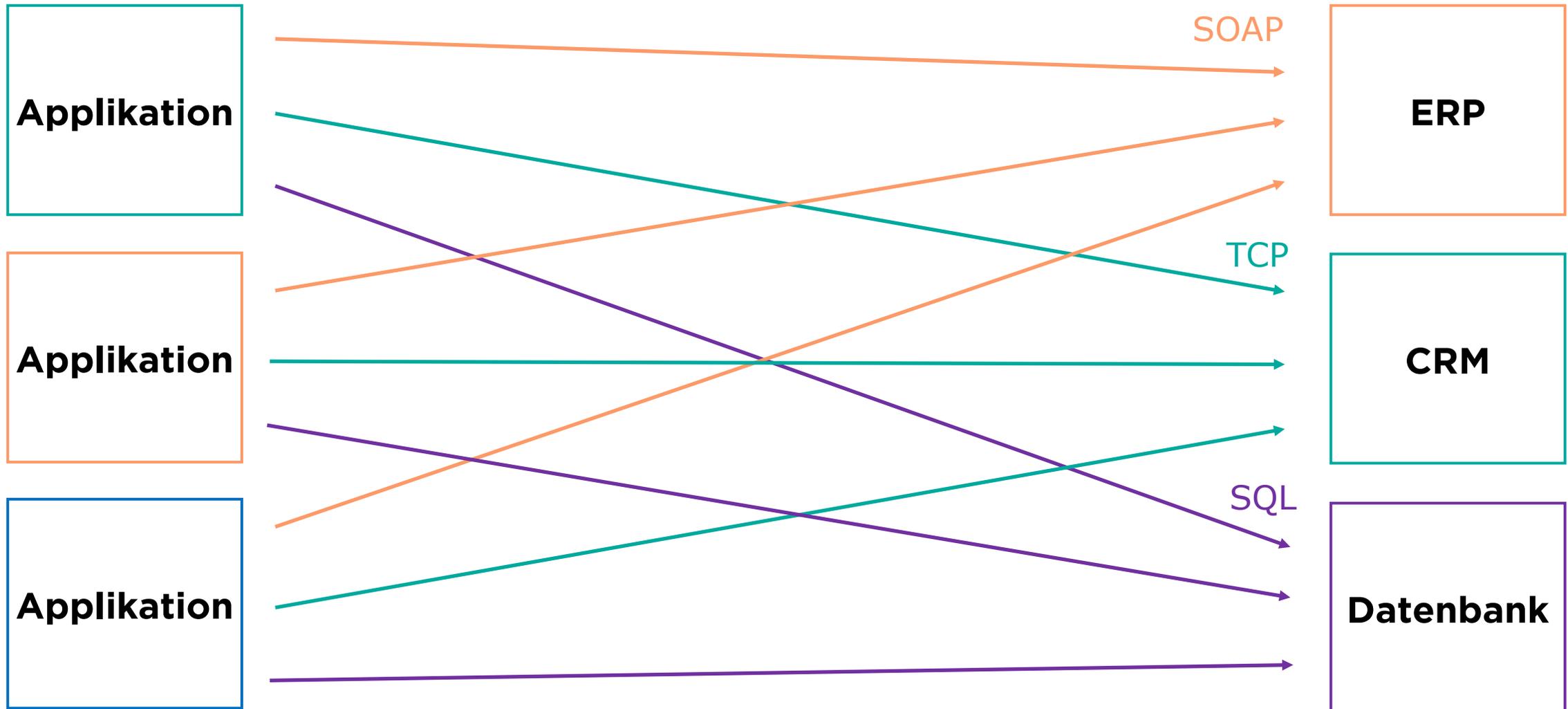
 **Unternehmensweit aktuelle und konsistente Daten und Metadaten**

 **Single source of truth für Geschäftsanwender, Analysten, Kunden und Anwendungen**

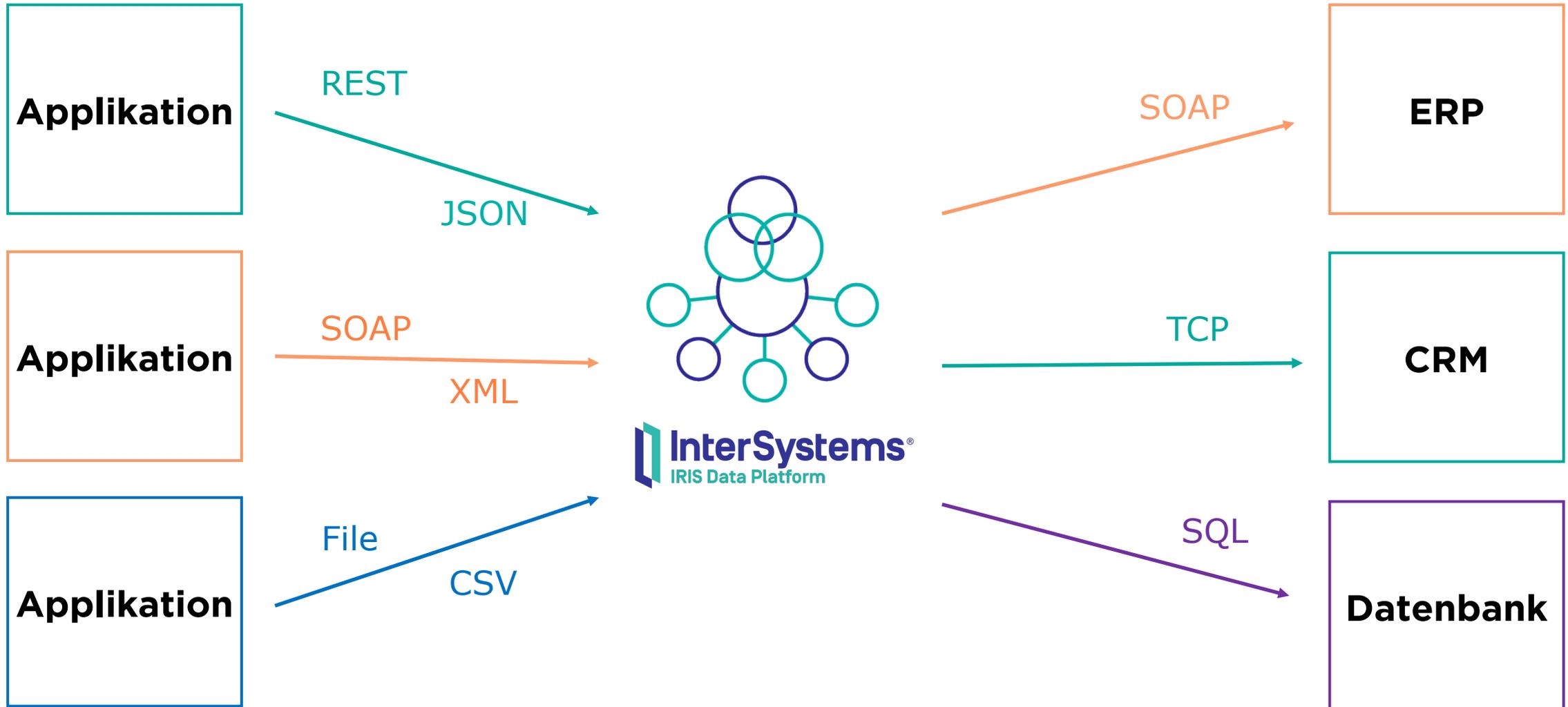
 **Ermöglicht interactive, dynamische Datenexploration**



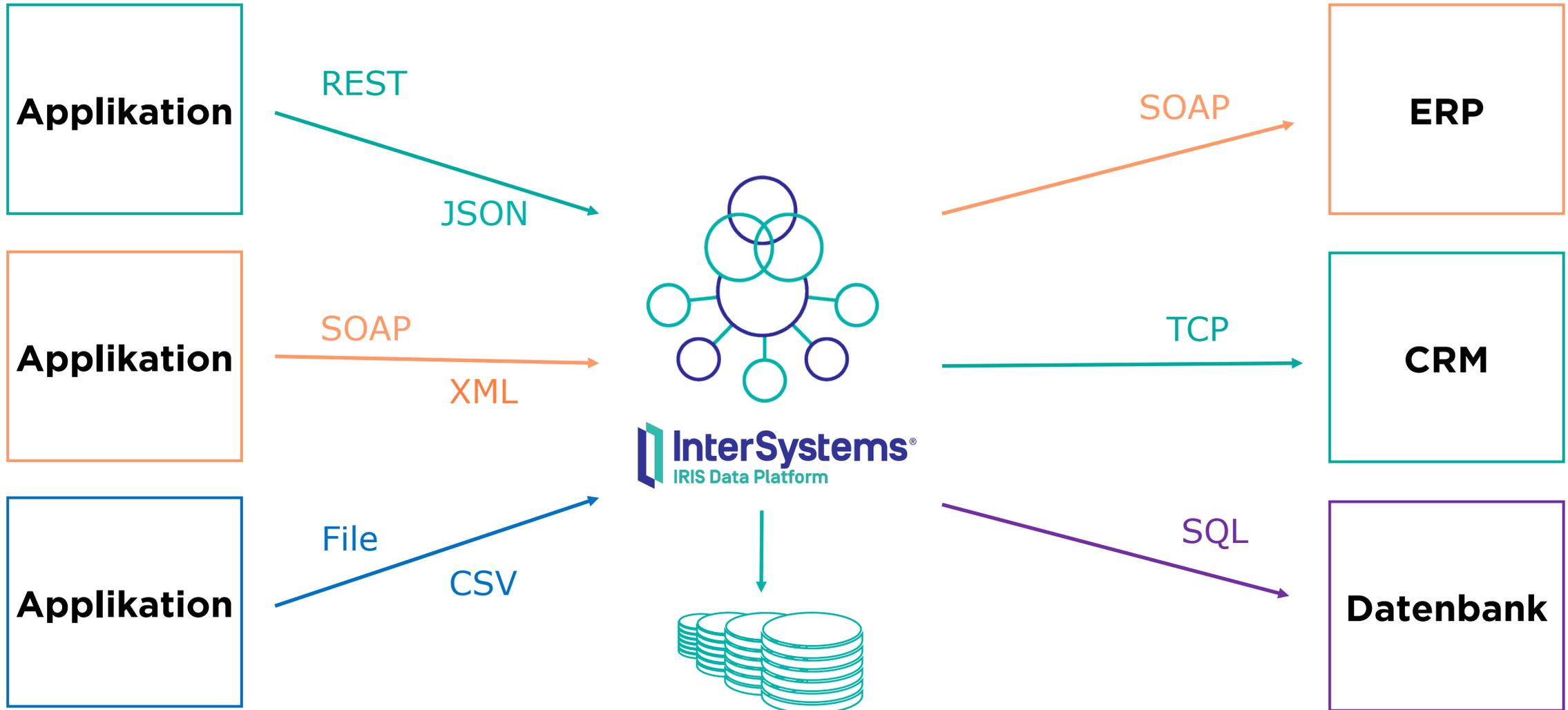
IRIS - Interaktion von Applikationen



IRIS - Interoperabilität



IRIS - Interoperabilität



IRIS Interoperabilität - Workflow

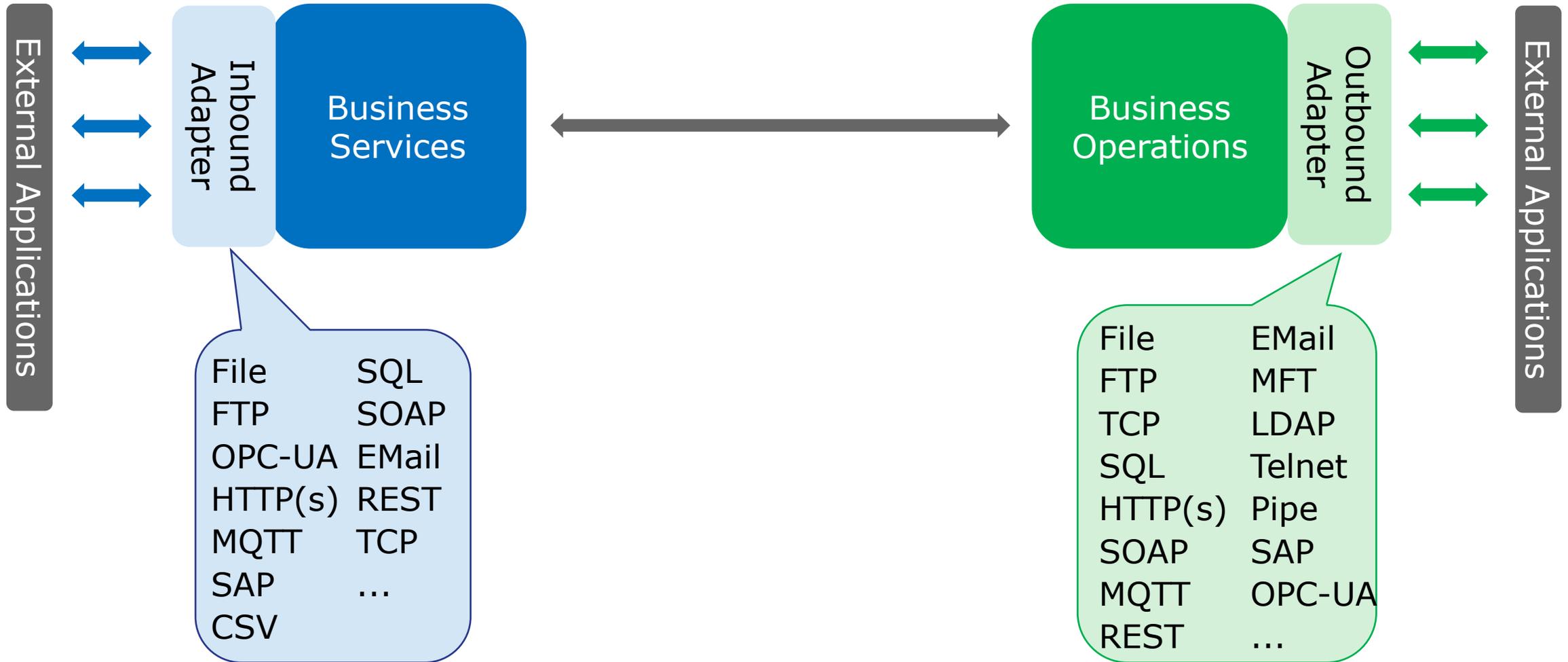


External Applications

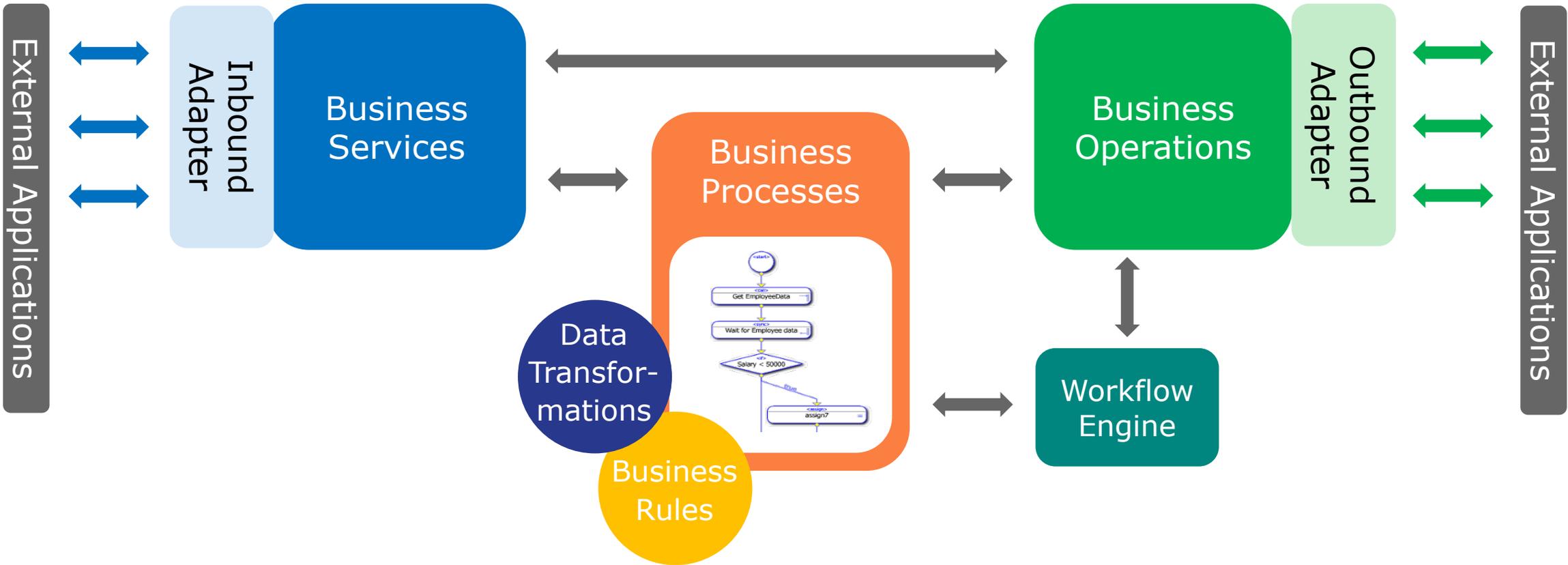


External Applications

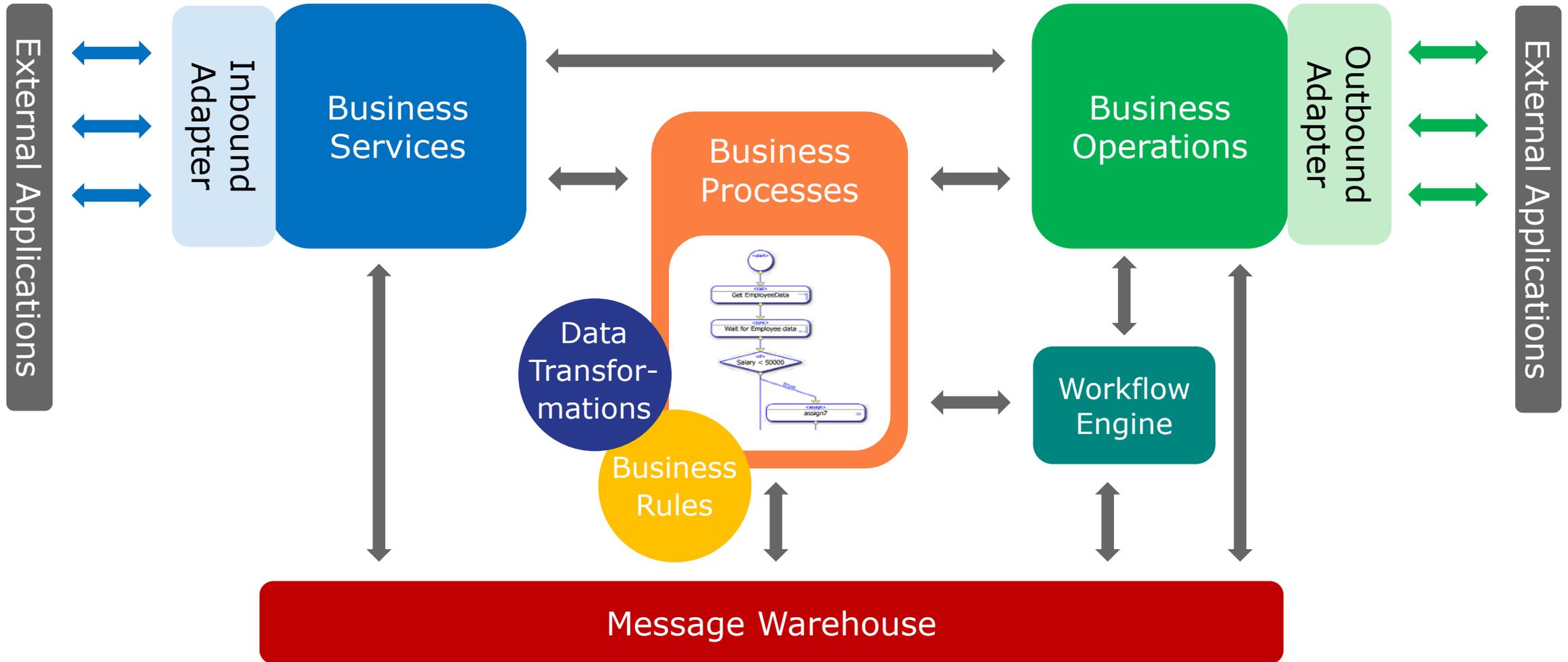
IRIS Interoperabilität - Workflow



IRIS Interoperabilität - Workflow



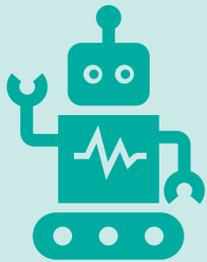
IRIS Interoperabilität - Workflow



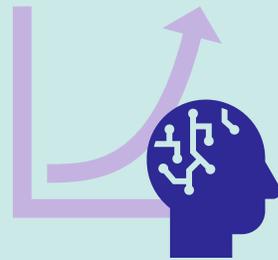
Was macht IRIS Smart?



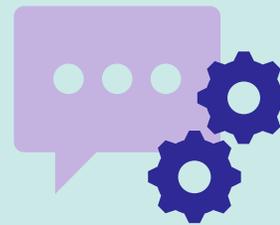
SMART FEATURES



Integrated ML

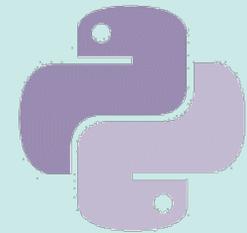


Advanced Analytics



Natural Language Processing

PYTHON ECOSYSTEM

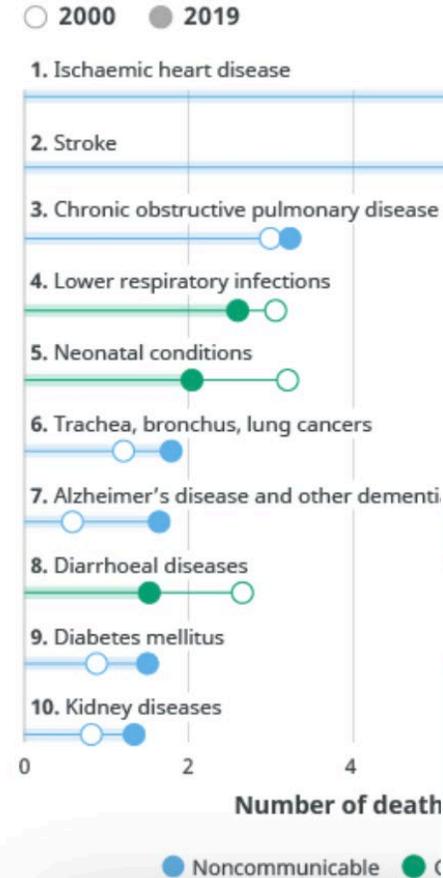


Embedded Python

Der Usecase

- Schlaganfall ist die zweithäufigste Todesursache weltweit (WHO)
- Sie leiten ein Krankenhaus und wollen sicherstellen, dass keiner Ihrer Patienten einen Schlaganfall erleidet, während er dort ist.

Leading causes of death globally



Rasu Shrestha MD MBA · 1st
Chief Innovation and Commercialization Officer, Executive Vice Pr...
3h · 🗨️

Every 40 seconds someone in the U.S. has a stroke and when treating a stroke, "Time is Brain." [Atrium Health](#) is one of the first health systems in Charlotte to provide tenecteplase (TNKase), a new clot-bustir ...see more

SIGNS OF A STROKE

CALL 911 Immediately!

B | **E** | **F** | **A** | **S** | **T**

LOSS OF BALANCE | VISION CHANGES | FACE DROOPING | ARM WEAKNESS | SPEECH DIFFICULTY | TERRIBLE HEADACHE TIME TO CALL 911

Now part of **ADVOCATEHEALTH**

Die Krankenhaus Anwendung

 George Costanza



Log Out

Stroke Risk

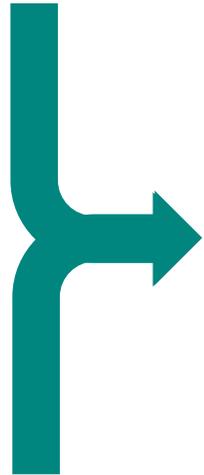
Gender	Age ↑	Hypertension	Heart Disease	Ever Married	Work Type	Residence Type	Average Glucose Level	BMI	Smoking Status	Probability Stroke
Female	43	0	0	No	Govt_job		101.35	32.2	never smoked	
Male	40	0	0	Yes	Private		64.84	26.6	never smoked	
Female	45	0	0	Yes	Private		176.48	24	formerly smoked	
Female	12	0	0	No	children		80.89	20.1	Unknown	
Female	29	0	0	No	Govt_job		57.02	43	formerly smoked	

Items per page: 5 1 - 5 of 981

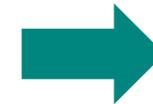


- Daten finden, die Patienteninformationen mit Schlaganfällen verknüpfen und ML-Modelle trainieren, um die Wahrscheinlichkeit zu berechnen.
- Hinzufügen einer Spalte für Wahrscheinlichkeit, um sie somit sortierbar zu machen.
- Medizinisches Team kann dies für die Bewertung der Behandlung und bessere Reaktion auf einen Schlaganfall nutzen.

Masterclass Architektur



InterSystems IRIS Data Platform



Angular
WebApp



Die Rohdaten

stroke_data_raw

id	67
gender	Female
date_birth	10/16/2003
hypertension	0
heart_disease	0
ever_married	No
work_type	Private
zip	98815
avg_glucose_level	92.97
height_in	59.1
weight_lb	0.0
smoking_status	formerly smoked
stroke	0

zip_codes

zip	98815
location	Seattle, WA
Residence_type	Urban

Agenda und Übungen

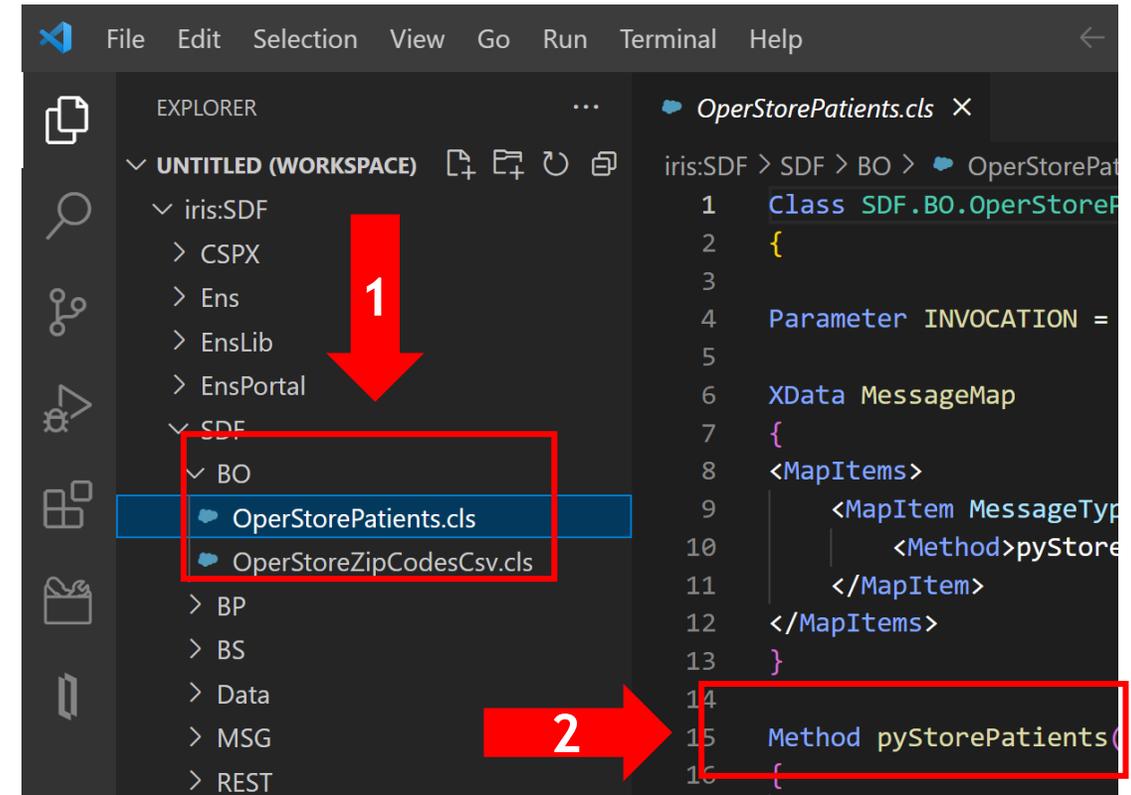
1. Erkunde die IRIS Interoperabilitätsschicht → Smart Data Fabric
2. Baue eine Operation um Patientendaten von einer externen Quelle einzulesen (MS SQL Server)
3. Einlesen von Postleitzahldaten aus einer anderen Quelle (CSV file)
4. Inputvariablen verändern mit Jupyter und SQL
5. Ein ML Modell mit IntegratedML trainieren
6. Einen REST-Service in IRIS von Angular aufrufen, um die Wahrscheinlichkeiten zu berechnen.



Übung 1:

Schreibe eine Operation um Patientendaten zu speichern

- Öffne Visual Studio Code in der VM
- Falls ein login erscheint drücke enter (Der UnknownUser hat kein Password)
- **1** Öffne die Klasse OperStorePatients in dem Paket HubBer.BO, falls es nicht schon offen ist.
- **2** Kopiere den Code in Method pyStorePatients (Beispielcode finden sie in der PDF auf dem Desktop)
- Drücke strg + S um den Code zu kompilieren



```
File Edit Selection View Go Run Terminal Help
EXPLORER
UNTITLED (WORKSPACE)
iris:SDF
  > CSPX
  > Ens
  > EnsLib
  > EnsPortal
  > SDF
    > BO
      OperStorePatients.cls
      OperStoreZipCodesCsv.cls
    > BP
    > BS
    > Data
    > MSG
    > REST
  > ...

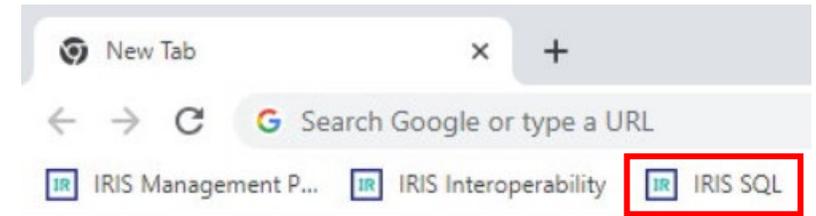
OperStorePatients.cls
iris:SDF > SDF > BO > OperStorePat
1 Class SDF.BO.OperStoreP
2 {
3
4 Parameter INVOCATION =
5
6 XData MessageMap
7 {
8 <MapItems>
9   <MapItem MessageType
10     <Method>pyStore
11   </MapItem>
12 </MapItems>
13 }
14 Method pyStorePatients(
15 {
```



Übung 2:

Anschauen der leeren Tabellen vor dem Importieren

- Öffne den IRIS SQL Explorer
- Filter nach der Tabelle HubBer_Data
- Drag and drop den Tabellennamen in das Eingabefeld and führe die Abfrage aus.
- Die beiden Tabellen sind leer. Die Daten werden wir in den nächsten Aufgaben importieren



SERVER Win11VM NameSpace SDF [wechseln](#) Benutzer UnknownUser Lizenziert für InterSystems IRIS Community Instanz IRIS

System > SQL

Filter gilt für Alle

System Schema

▼ Tabellen

- > SDF_Data.stroke_data_raw
- > SDF_Data.zip_codes

▼ Views

Assistenten >> Aktionen >> Tabelle öffnen Werkzeuge >> Dokumentation >>

Katalogdetails Abfrage ausführen Durchsuchen SQL-Statements

Ausführen Plan anzeigen Verlauf anzeigen Abfrage-Generator Display-Modus ▼ Max 1000 mehr

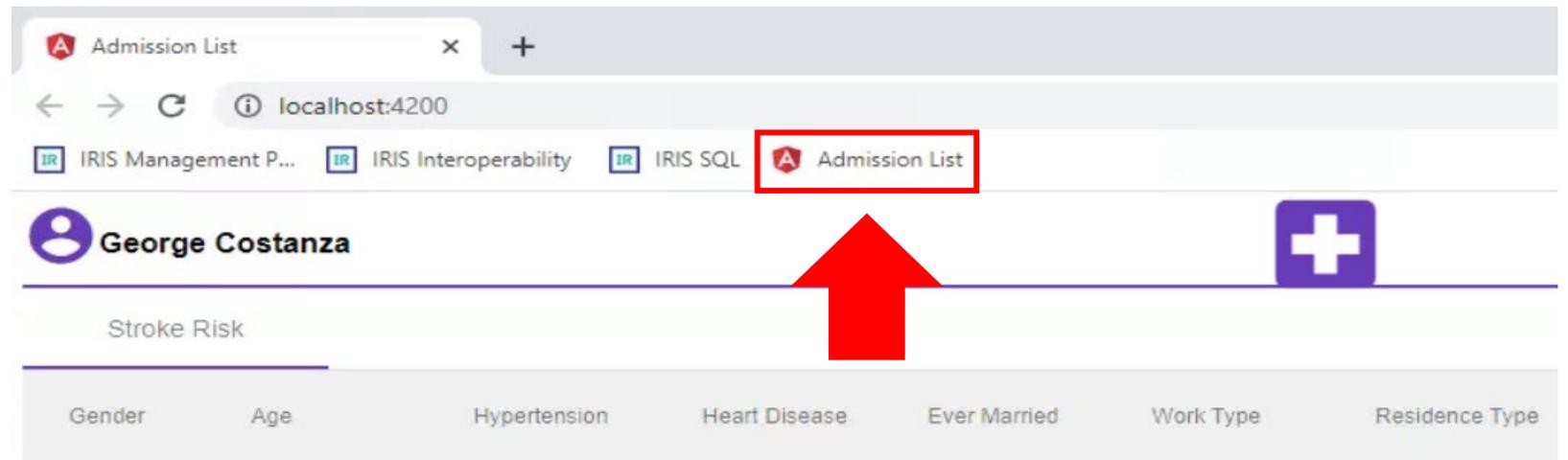
```
SELECT
id, gender, date_birth, hypertension, heart_disease, ever_married, work_type, zip,
avg_glucose_level, height_in, weight_lb, smoking_status, stroke
FROM SDF_Data.stroke_data_raw
```



Notiz: Anschauen der Angular Oberfläche.

Es sollten keine Daten angezeigt werden. Wir werden jetzt Daten importieren...

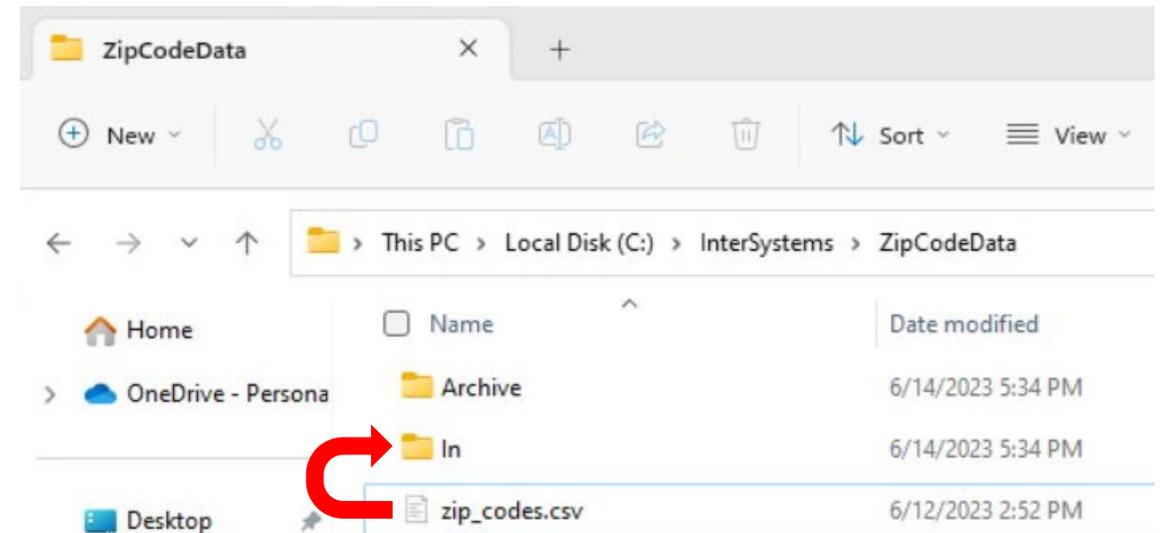
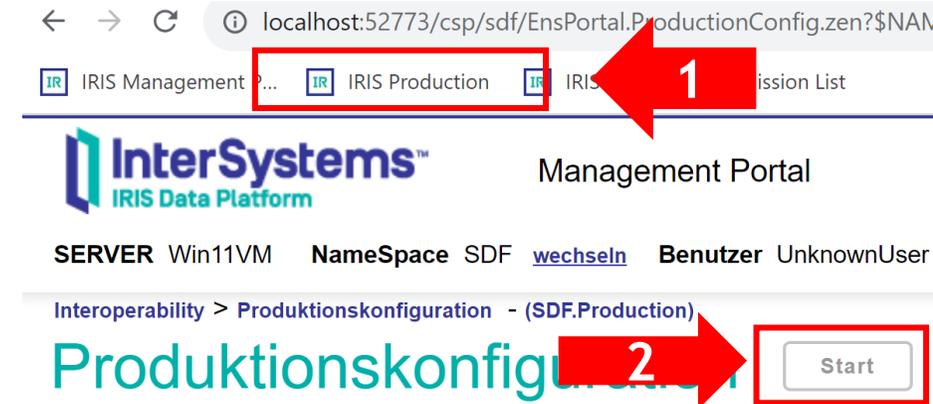
<http://localhost:4200> (Als Lesezeichen in Chrome:)



Übung 3:

Importieren der Postleitzahldaten in IRIS

1. Öffne Production in IRIS über Chrome:
2. Starte die Production durch klicken auf Start
3. Durchsuche das Laufwerk nach C:\InterSystems\ZipCodeData
4. 3.1 Kopiere die zip_codes.csv in den In Ordner, und schaue wie es automatisch in IRIS importiert wird



Übung 4:

Importieren der Patientendaten in IRIS

- Starte den Service ServStartPatientsImport durch doppelklicken auf den Service und dann aktivieren klicken.

SERVER Win11VM Namespace SDF [wechseln](#) Benutzer UnknownUser Lizenziert für

Interoperability > Produktionskonfiguration - (SDF.Production)

Produktionskonfiguration

Produktion wird ausgeführt Kategorie:

Dienste	Prozesse
<input type="radio"/> ServStartIntegratedML	<input checked="" type="radio"/> ProcIntegratedML
<input type="radio"/> ServStartPatientsImport	<input checked="" type="radio"/> ProcPatients
<input checked="" type="radio"/> ServZipCodesCsvIn	<input checked="" type="radio"/> ProcZipCodesCsv



Übung 5:

Die Tabellen nochmal anschauen

- Gehe in den IRIS SQL Explorer
- Filter nach der Tabelle HubBer_Data, Drag and drop die Tabelle in das Eingabefenster and führe die Abfrage aus um die importierten Daten zu sehen. Es sollte so aussehen:

HubBer_Data.stroke_data_raw

id	gender	date_birth	hypertension	heart_disease	ever_married	work_type	zip	avg_glucose_level	height_in	weight_lb
954	Male	1/9/2002	0	0	No	Private	99006	103.939999999999772	60.200001000000000317	120.1099999999999943
2182	Female	10/22/1940	1	0	Yes	Self-employed	98001	91.01999700000003596	61.7999899999999682	178.7400000000000909
3113	Female	10/23/1987	0	0	No	Private	98274	80.20999899999996271	60.200001000000000317	143.3100000000000227
4117	Female	5/29/1964	0	0	Yes	Self-employed	98446	81.76999700000003596	66.5	137.1299999999999546
5121	Male	12/8/1990	0	0	Yes	Private	98279	96.8399959999999298	60.59999799999999366	110.2199999999999886
6049	Female	3/23/2015	0	0	No	children	98661	73.690002000000006887	44.5	69.8599999999999431
7047	Female	4/1/1989	0	0	Yes	Private	99350	69.72000099999996338	67.69999699999996206	257.5199999999998181
7990	Female	9/23/1996	0	0	Yes	Private	99218	84.08000199999993246	70.900002000000000634	175.1899999999999772
8964	Female	10/13/1986	0	0	No	Private	98221	94.37000299999996973	59.400002000000000634	191.2199999999999886
10323	Female	4/13/1954	0	0	Yes	Private	98685	112.7699999999999602	66.09999799999999366	141.08000000000001251
11259	Female	1/22/1967	0	0	Yes	Private	98365	227.67999000000000364	74	0
12228	Male	6/11/2007	0	0	No	children	98037	97.97000099999996338	61.400002000000000634	131.3899999999998636
13286	Male	10/24/2017	0	0	No	children	99357	81	39	44.7899999999999147
14287	Female	7/30/1988	0	0	Yes	Private	99185	68.660004000000000701	66.09999799999999366	140.46000000000000796

HubBer_Data.zip_codes

zip	location	Residence_type
98001	Rockford, WA	Rural
98002	White Swan, WA	Urban
98003	Riverside, WA	Rural
98004	Touchet, WA	Rural
98005	Naches, WA	Rural
98006	Cusick, WA	Rural
98007	Tieton, WA	Urban
98008	Ocean Park, WA	Rural
98010	Spokane, WA	Urban
98011	Coulee Dam, WA	Rural
98012	Rice, WA	Rural
98014	Spokane, WA	Urban
98019	Steilacoom, WA	Urban
98020	Onalaska, WA	Rural

Diese Übung beinhaltet folgendes:

1. Hinzufügen der Wohnorttypen jedes Patienten (urban or rural), in dem man die Postleitzahlen beider Tabellen `stroke_data_raw` und `zip_codes` vereint;
2. Berechne das Alter jedes Patienten mit dem Geburtstag, unter der Annahme, dass die Daten am 12/31/2020 gesammelt wurden;
3. Berechne das BMI jedes Patienten basierend auf Gewicht und Größe, Berücksichtigung fehlender Daten, mittels der folgenden Formel:

$$BMI = \frac{weight(lb)}{height(in)^2} \times 703$$



Arbeitsblatt

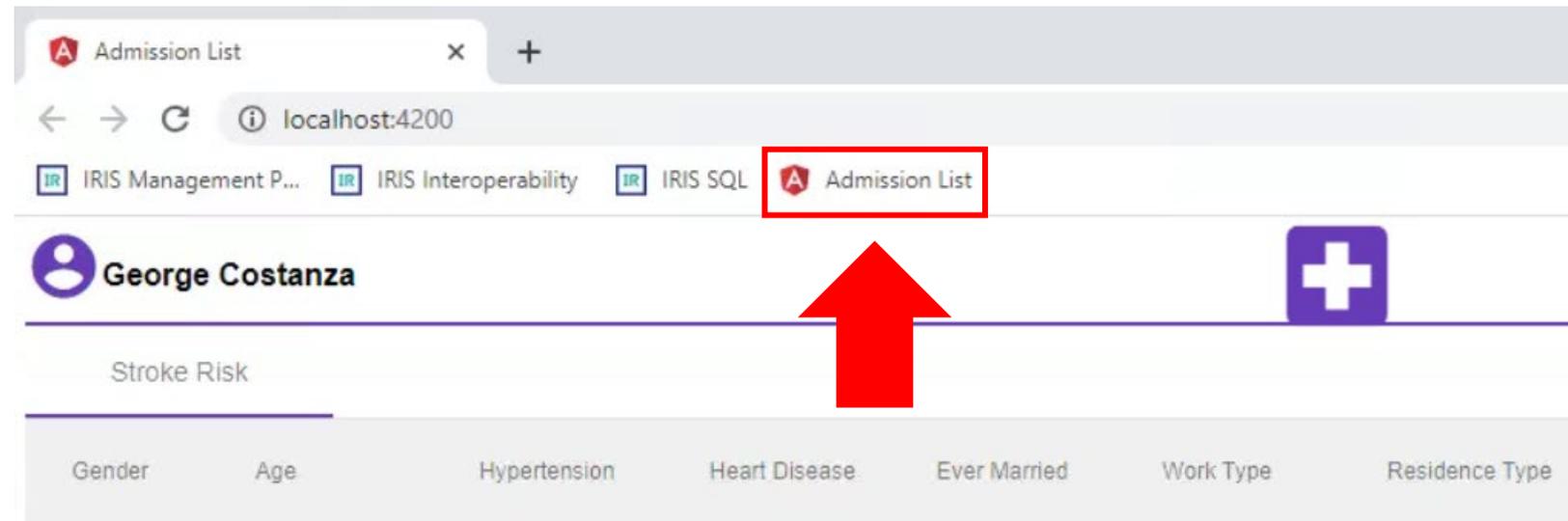
Folge den Anweisungen in dem ersten Jupyter-
Notebook
um die Daten vorzubereiten.



Notiz: Anschauen der Angular Oberfläche.

Die bearbeiteten Daten sollten jetzt da sein.

<http://localhost:4200> (Lesezeichen in Chrome):



Machine Learning mag leicht vergleichbare Werte



Wir wollen unsere Daten auf sinnvolle Weise kombinieren und anreichern.

Wie können wir das tun?

Vergleichbare Werte finden..

stroke_data_raw

id	67
gender	Female
date_birth	10/16/2003
hypertension	0
heart_disease	0
ever_married	No
work_type	Private
zip	98815
avg_glucose_level	92.97
height_in	59.1
weight_lb	0.0
smoking_status	formerly smoked
stroke	0

zip_codes

zip	98815
location	Seattle, WA
Residence_type	Urban



Machine Learning likes square data

X = y

id ^	gender ^	date_birth ^	hypertension ^	heart_disease ^	ever_married ^	work_type ^	zip ^	stroke ^
67	Female	10/16/2003	0	0	No	Private	98815	0
77	Female	4/5/2007	0	0	No	children	98671	0
84	Male	5/17/1965	0	0	Yes	Private	99126	0
91	Female	3/29/1978	0	0	No	Private	98382	0
99	Female	12/24/1989	0	0	No	Private	98323	0
121	Female	4/1/1982	0	0	Yes	Private	98122	0
129	Female	8/23/1996	0	0	No	Private	98465	0
132	Female	1/21/1940	0	0	Yes	Govt_job	99016	0



Machine Learning likes square data

X = y

id ^	gender ^	date_birth ^	hypertension ^	heart_disease ^	ever_married ^	work_type ^	zip ^	stroke ^
67	Female	10/16/2003	0	0	No	Private	98815	0
77	Female	4/5/2007	0	0	No	children	98671	0
84	Male	5/17/1965	0	0	Yes	Private	99126	0
91	Female	3/29/1978	0	0	No	Private	98382	0
99	Female	12/24/1989	0	0	No	Private	98323	0
121	Female	4/1/1982	0	0	Yes	Private	98122	0
129	Female	8/23/1996	0	0	No	Private	98465	0
132	Female	1/21/1940	0	0	Yes	Govt_job	99016	0

Jede Zeile ist einzelne Vorhersage



Machine Learning likes square data

$f(x)$

= y

id ^	gender ^	date_birth ^	hypertension ^	heart_disease ^	ever_married ^	work_type ^	zip ^	stroke ^
67	Female	10/16/2003	0	0	No	Private	98815	0
77	Female	4/5/2007	0	0	No	children	98671	0
84	Male	5/17/1965	0	0	Yes	Private	99126	0
91	Female	3/29/1978	0	0	No	Private	98382	0
99	Female	12/24/1989	0	0	No	Private	98323	0
121	Female	4/1/1982	0	0	Yes	Private	98122	0
129	Female	8/23/1996	0	0	No	Private	98465	0
132	Female	1/21/1940	0	0	Yes	Govt_job	99016	0

Jede Zeile ist einzelne Vorhersage



Machine Learning likes square data

$f(x)$

= y

id ^	gender ^	date_birth ^	hypertension ^	heart_disease ^	ever_married ^	work_type ^	zip ^	stroke ^
67	Female	10/16/2003	0	0	No	Private	98815	0
77	Female	4/5/2007	0	0	No	children	98671	0
84	Male	5/17/1965	0	0	Yes	Private	99126	0
91	Female	3/29/1978	0	0	No	Private	98323	0
99	Female	12/24/1989	0	0	No	Private	98323	0
121	Female	4/1/1982	0	0	Yes	Private	98122	0
129	Female	8/23/1996	0	0	No	Private	98465	0
132	Female	1/21/1940	0	0	Yes	Govt_job	99016	0

Jede Spalte ist eine Serie von Daten



IntegratedML

- Ermöglicht es, ein ML-Modell mit nur vier SQL-Anweisungen zu erstellen.
- Bringt ML zu den Daten, anstatt die Daten auf eine völlig andere Plattform zu verlagern.
- Nutzung von AutoML-Tools, die den Aufwand für die Erstellung von ML-Modellen verringern.
- Es ist sehr einfach, Ihr Modell wieder in Ihre Anwendung einzubinden - auch hier verwenden Sie einfach SQL.



IntegratedML

Dieser Provider wird Folgendes ausführen:

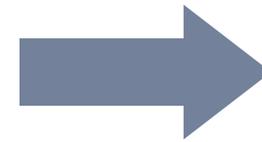
Feature Transformation (Umgang mit Nullen, kodierten Kategoriefeldern, tokenisiertem Text, standardisierten Zahlenwerten usw.)

- Mehrere Modelltypen ausprobieren (boosted trees, logistic regression, neural networks, etc.)
- Zusammenstellung der Siegermodelle aus der Rangliste zu einem einzigen Hochleistungsmodell
- Alles mit dem Komfort unserer SQL-Tools



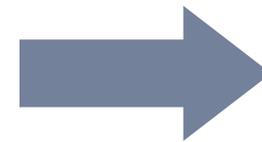
InterSystems AutoML - das SQL-Interface

```
CREATE MODEL <model name>  
PREDICTING(<label>)  
FROM <table/view>
```



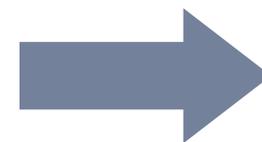
ML Model
Definition

```
TRAIN MODEL <model name>
```



Trained
ML Model

```
SELECT PREDICT(<model name>)  
FROM <table/view/record>
```



Predicted
Result



Arbeitsblatt:

Folge den Anweisungen im zweiten Jupyter-
Notebook
um das ML Model zu trainieren.



So sehen die Metriken aus:

Actual Values

Stroke

Not Stroke

Predicted Values

Stroke

42

254

296

Not Stroke

6

689

695

48

943

991

$$\text{Accuracy} = (42 + 689) / 911 = 74\%$$

$$\text{Recall} = 42 / 48 = 88\%$$

$$\text{Precision} = 42 / 296 = 14\%$$



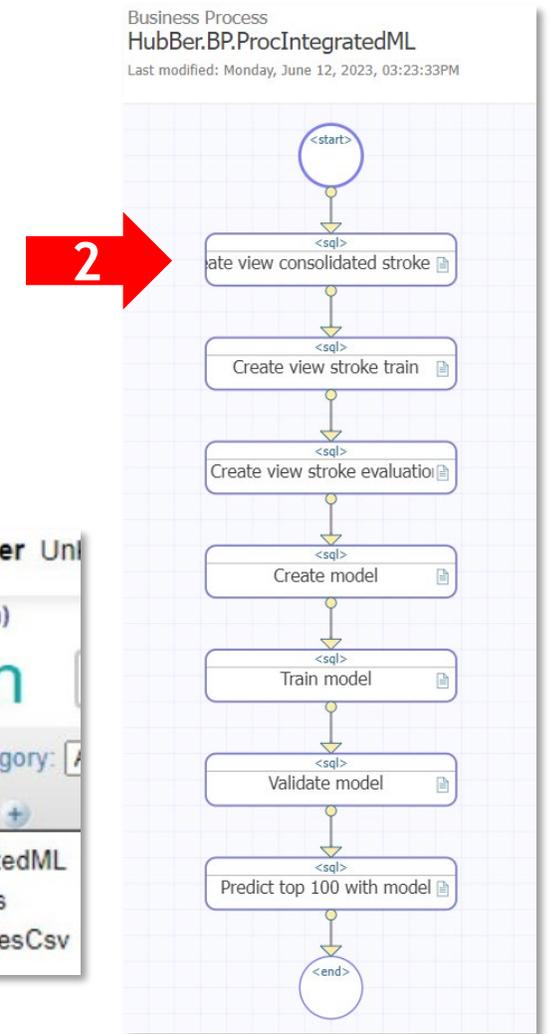
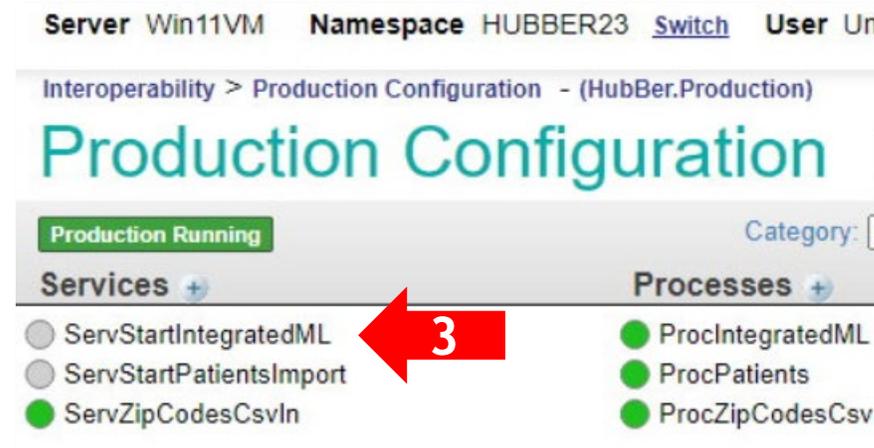
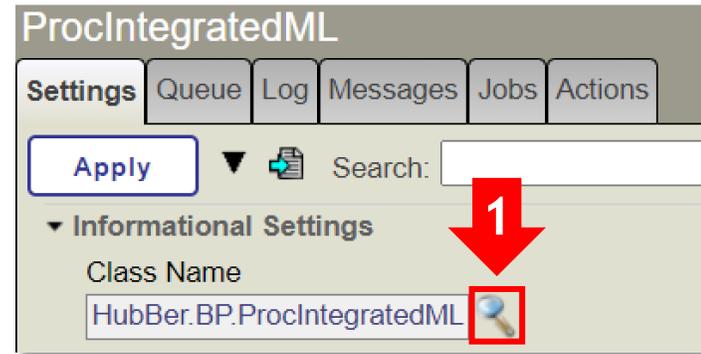
Accuracy als Metrik

- Accuracy funktioniert nicht für “Nadel in Heuhaufen” Probleme
- Nur 5 % der Patienten in unseren Daten haben einen Schlaganfall
- Außerdem ist es manchmal besser, sich in einer Richtung zu irren als in der anderen.



Übung 6 (nach Jupyter): Den IntegratedML Prozess starten

- Öffne den Prozess ServStartIntegratedML mit der Lupe im Seitenfenster
- Durch Klicken auf die Aktivitäten, sieht man die Gleichen SQL Abfragen wie im Jupyter-Notebook
- Starte den Service ServStartIntegratedML durch Doppelklick und aktivieren klicken



Application Architecture

- Angular UI ruft den REST service am IRIS backend auf
- REST service führt Abfrage gegen die Tabelle aus
 - Verwendung der PROBABILITY-Anweisung als Teil von Native SQL zum Aufrufen des IntegratedML-Modells
 - REST-Dienst gibt JSON-Ergebnis zur Anzeige an die Benutzeroberfläche zurück



Übung 6:

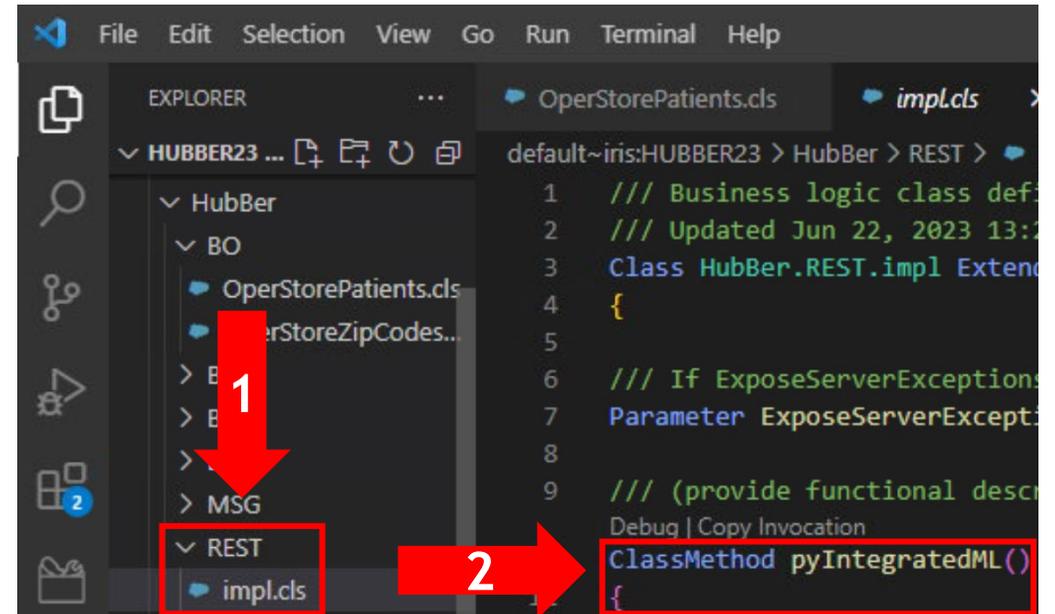
Bearbeiten der REST API um die Wahrscheinlichkeiten eines Schlaganfalls zu berechnen

- Wechsle zu Visual Studio Code auf der VM
- Öffne die Klasse impl in dem Paket SDF.REST, falls es nicht schon offen ist
- Füge folgenden Code in Linie 34 hinzu:

```
'probability_stroke':ROUND(PROBABILITY(StrokeModel USE StrokeModelTrained),5)
```

- Den Code für die ganze Methode befindet sich auch auf dem Desktop als text file on → “Übung 7 method pyIntegratedML.txt”)

Screenshot auf der nächsten Slide

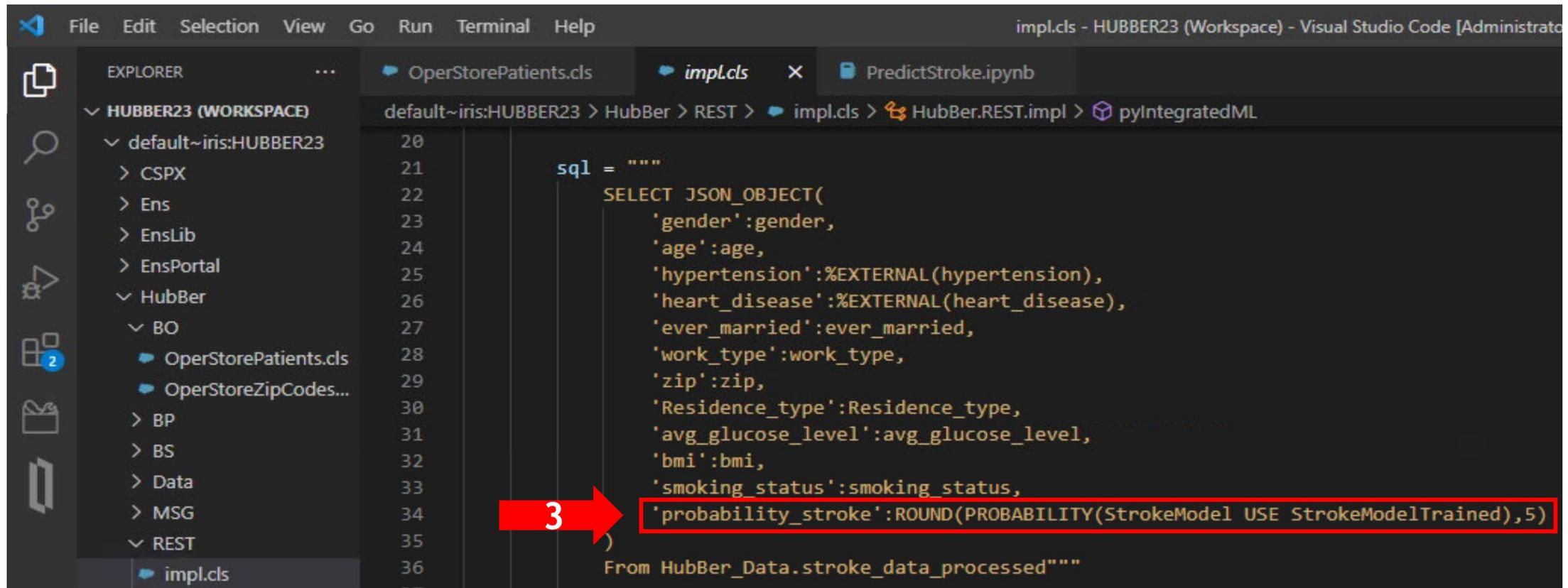


```
File Edit Selection View Go Run Terminal Help
EXPLORER
HUBBER23 ...
  HubBer
    BO
      OperStorePatients.cls
      OperStoreZipCodes..
    > B
    > B
    > B
    > MSG
    REST
      impl.cls
OperStorePatients.cls
impl.cs
default~iris:HUBBER23 > HubBer > REST >
1  /// Business logic class def
2  /// Updated Jun 22, 2023 13:2
3  Class HubBer.REST.impl Extens
4  {
5
6  /// If ExposeServerExceptions
7  Parameter ExposeServerExcepti
8
9  /// (provide functional desc
Debug | Copy Invocation
ClassMethod pyIntegratedML()
{
```



Übung 6:

Bearbeiten der REST API um die Wahrscheinlichkeiten eines Schlaganfalls zu berechnen



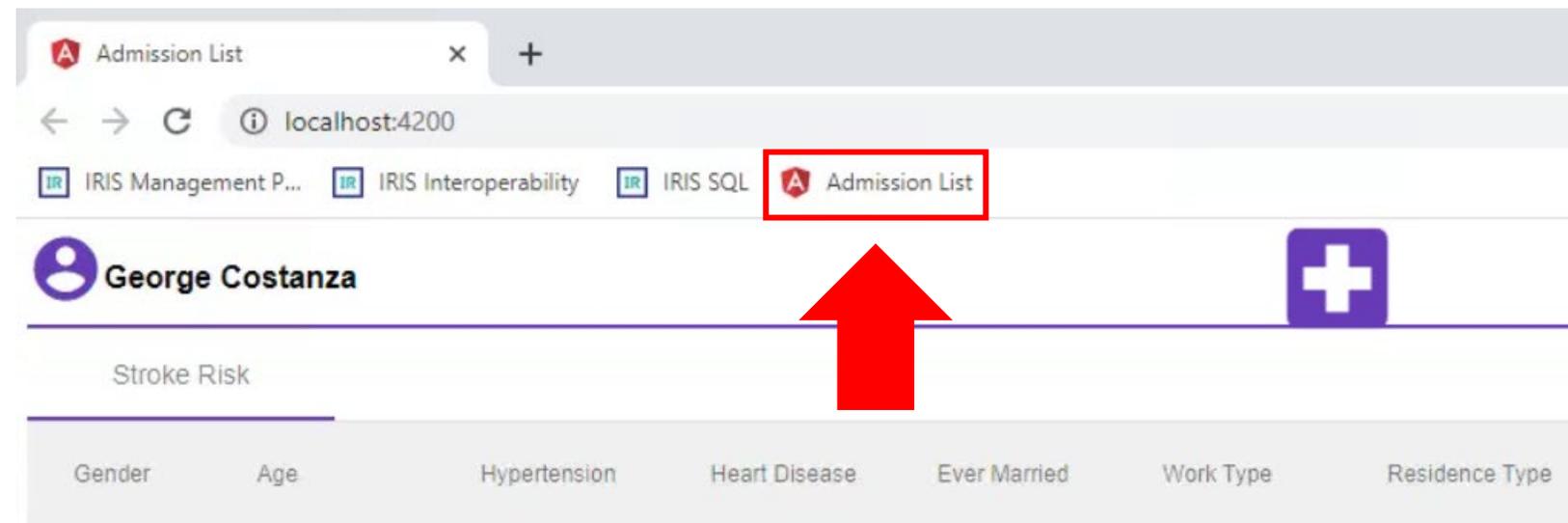
```
File Edit Selection View Go Run Terminal Help impl.cls - HUBBER23 (Workspace) - Visual Studio Code [Administrato]
EXPLORER
HUBBER23 (WORKSPACE)
  default~iris:HUBBER23
    CSPX
    Ens
    EnsLib
    EnsPortal
    HubBer
      BO
        OperStorePatients.cls
        OperStoreZipCodes...
      BP
      BS
      Data
      MSG
    REST
      impl.cls
      OperStorePatients.cls
      impl.cls
      PredictStroke.ipynb
default~iris:HUBBER23 > HubBer > REST > impl.cls > HubBer.REST.impl > pyIntegratedML
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
sql = """
    SELECT JSON_OBJECT(
        'gender':gender,
        'age':age,
        'hypertension':%EXTERNAL(hypertension),
        'heart_disease':%EXTERNAL(heart_disease),
        'ever_married':ever_married,
        'work_type':work_type,
        'zip':zip,
        'Residence_type':Residence_type,
        'avg_glucose_level':avg_glucose_level,
        'bmi':bmi,
        'smoking_status':smoking_status,
        'probability_stroke':ROUND(PROBABILITY(StrokeModel USE StrokeModelTrained),5)
    )
    From HubBer_Data.stroke_data_processed"""
```



Notiz: Anschauen der Angular Oberfläche.

Es sollten nun die verarbeiteten Daten mit den berechneten Wahrscheinlichkeiten angezeigt werden.

<http://localhost:4200> (Lesezeichen in Chrome):



Rückblick

ML ist ein mächtiger Ansatz, um Anwendungen intelligenter zu machen
InterSystems macht es sehr einfach, die Brücke zwischen den ML-Welten
und den Anwendungs- und Datenwelten zu bauen. Wir "bringen ML zu
den Daten".

Technologien die verwendet wurden:

- InterSystems IRIS Data Platform
- IRIS IntegratedML
- Python-Jupyter easily using IRIS data
- Embedded Python for the Operation



Vielen Dank.

Jannis Stegmann

Sales Engineer

jannis.stegmann@intersystems.com

Mobil: +49 174 2125082

Felix Vetter

Sales Engineer Intern

felix.vetter@intersystems.com

Mobil: +49 174 4797827