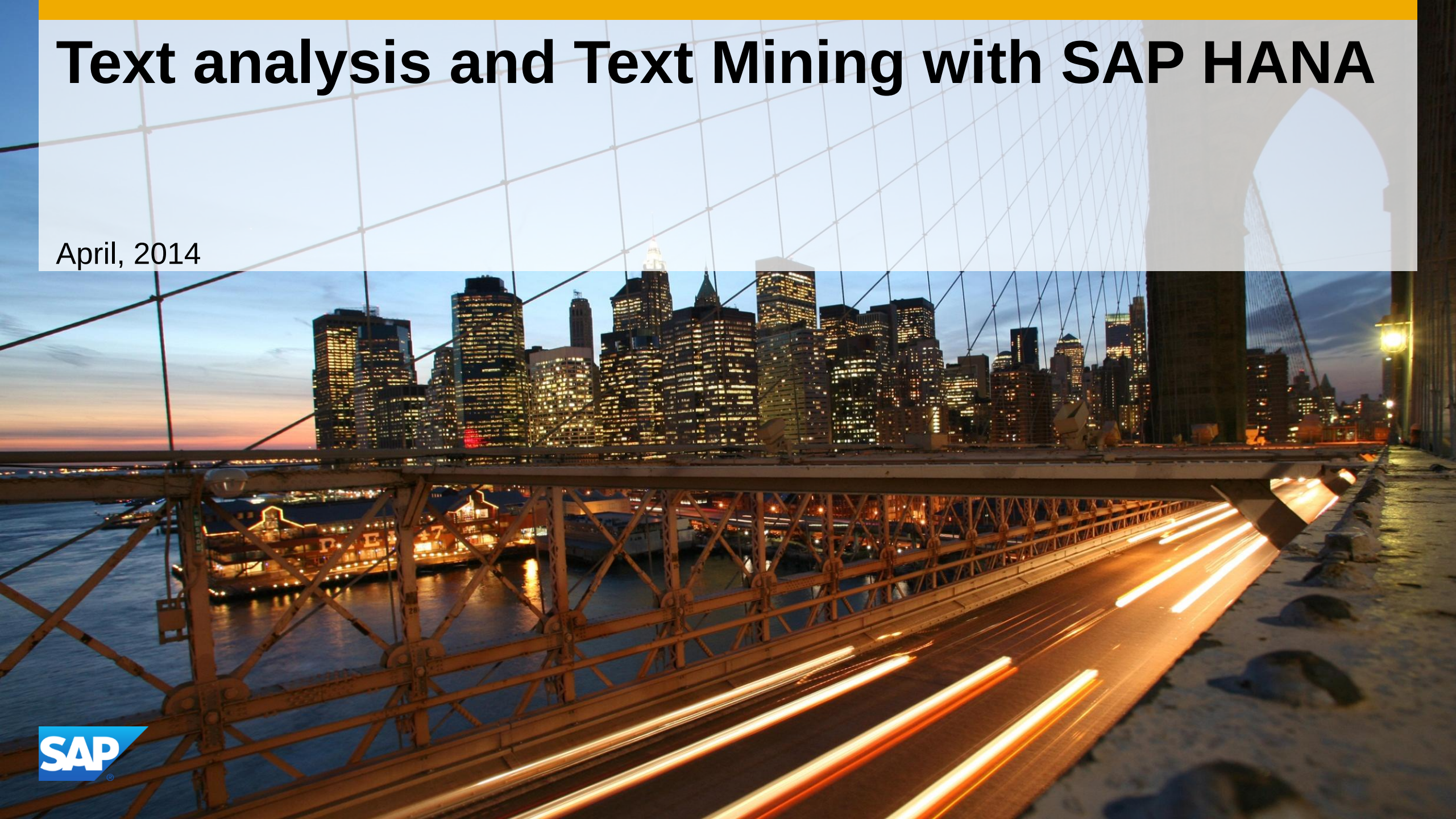
Text analysis and Text Mining with SAP HANA

April, 2014



Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

What vs. Why

It’s often said that

- Structured data tells us “what”
- Unstructured data tells us “Why”

Hidden Value in Text

80% of enterprise-relevant information originates in “unstructured” data:

- Blogs, forum postings, social media
- Email, contact-center notes
- Surveys, warranty claims

SAP HANA Text Analysis Overview

SAP HANA

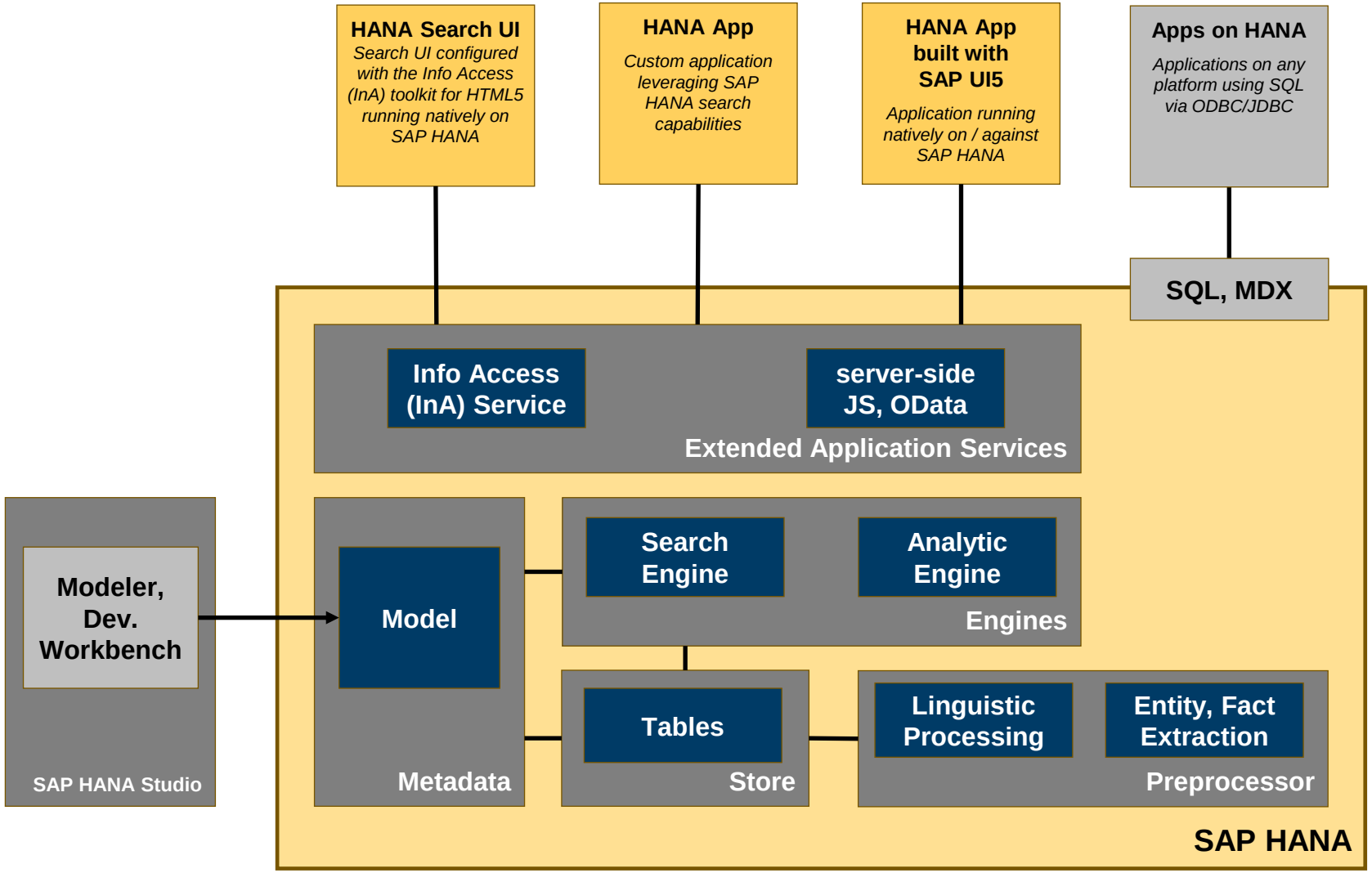
In-Memory Data Platform for Real-Time Analytics & Applications

SAP HANA

Information Composer and Modeling Studio	Text Search & Text Analysis	Spatial Analysis
Planning and Calculation Engine	Predictive Analysis & Business Function Libraries	R and Hadoop Integration
Real-time Replication Services	In-Memory Database	Data Services

SAP HANA

Full-Text Search Architecture



SAP HANA Info Access (InA) toolkit,
SInA API for JavaScript

Full-Text search via SQL

Modeling,
Search Rule Sets

Full-Text Indexing, Text Analysis

Steps to Build a Search Based Application

Configure/ Develop App	Use InA toolkit to define layout and data for search UIs Use SInA API and SAP UI5 to develop Search Applications
Create Model	Use HANA Studio to define search models and to specify search behavior
Run Text Analysis	Use HANA Studio to extract salient information from text
Create Full-Text Index	Use HANA Studio to create full-text indexes to enable full-text search
Get Data	Load data into HANA tables

Full-Text Indexing and Text Analysis

SAP HANA
Full-Text Indexing

A full-text index enables full text search

When a FULL-TEXT INDEX is created these steps are executed

- File filtering
  - Converting binary file types like .pdf, .ppt into plain text
- Linguistic Analysis
  - Tokenization - decompose word sequence, e.g. “the quick brown fox” -> “the” “quick” “brown” “fox”
  - Stemming - reduction of tokens to linguistic base form, e.g. houses -> house; ran -> run
  - Part-of-Speech identification, e.g. quick: Adjective; houses: Noun-Plural

Full-Text index is “attached” to the table column

SAP HANA
Text Analysis

HANA supports in-database Text Analysis

Text Analysis is an optional process “on top of” full-text indexing

Text Analysis results are stored in a table

- Multiple text analysis options
  - Linguistic markup, e.g. tokens, stems, POS tags
  - Entity extraction, e.g. persons, locations, dates etc.
  - “Voice of Customer” fact extraction, e.g. sentiments, requests, topics etc
- Language support
  - Up to 31 langauges

Search “Hello World!”

```
CREATE COLUMN TABLE ARTICLES (
        ID              INTEGER PRIMARY KEY,
        ARTICLE         NCLOB
);

INSERT INTO ARTICLES VALUES (1,'yesterday, the congress [..]');

CREATE FULLTEXT INDEX MY_INDEX ON ARTICLES(ARTICLE);

SELECT * FROM ARTICLES WHERE CONTAINS(ARTICLE,'congress');
```

More information: SAP HANA Database - SQL Reference, Fulltext Search WIKI

Tokenize

CREATE FULLTEXT INDEX <index_name> ON <tableref> ‘(’ <column_name> ‘)’[<fulltext_parameter_list>]
TEXT ANALYSIS on

TA_RULE	TA_COUNTER	TA_TOKEN	TA_LANGUAGE	TA_TYPE	TA_NORMALIZED
LXP	2	是	zh	verb	是
LXP	6	翻译	zh	verb	翻译
LXP	11	一本	zh	number	一本
LXP	9	著作	zh	noun	著作
LXP	3	中科院	zh	noun	中科院
LXP	4	心理	zh	noun	心理
LXP	5	所组织	zh	verb	所组织
LXP	8	一套	zh	number	一套

TEXT ANALYSIS ON CONFIGURATION '<NAME OF TEXT ANALYSIS CONFIGURATION>‘

LINGANALYSIS_BASIC：tokenize
LINGANALYSIS_STEMS：tokenize, stemming
LINGANALYSIS_FULL：tokenize, stemming, POS tags
EXTRACTION_CORE：person, location, orgnization…

Search Models

SAP HANA
Search Models

You use SAP HANA Studio to create search models

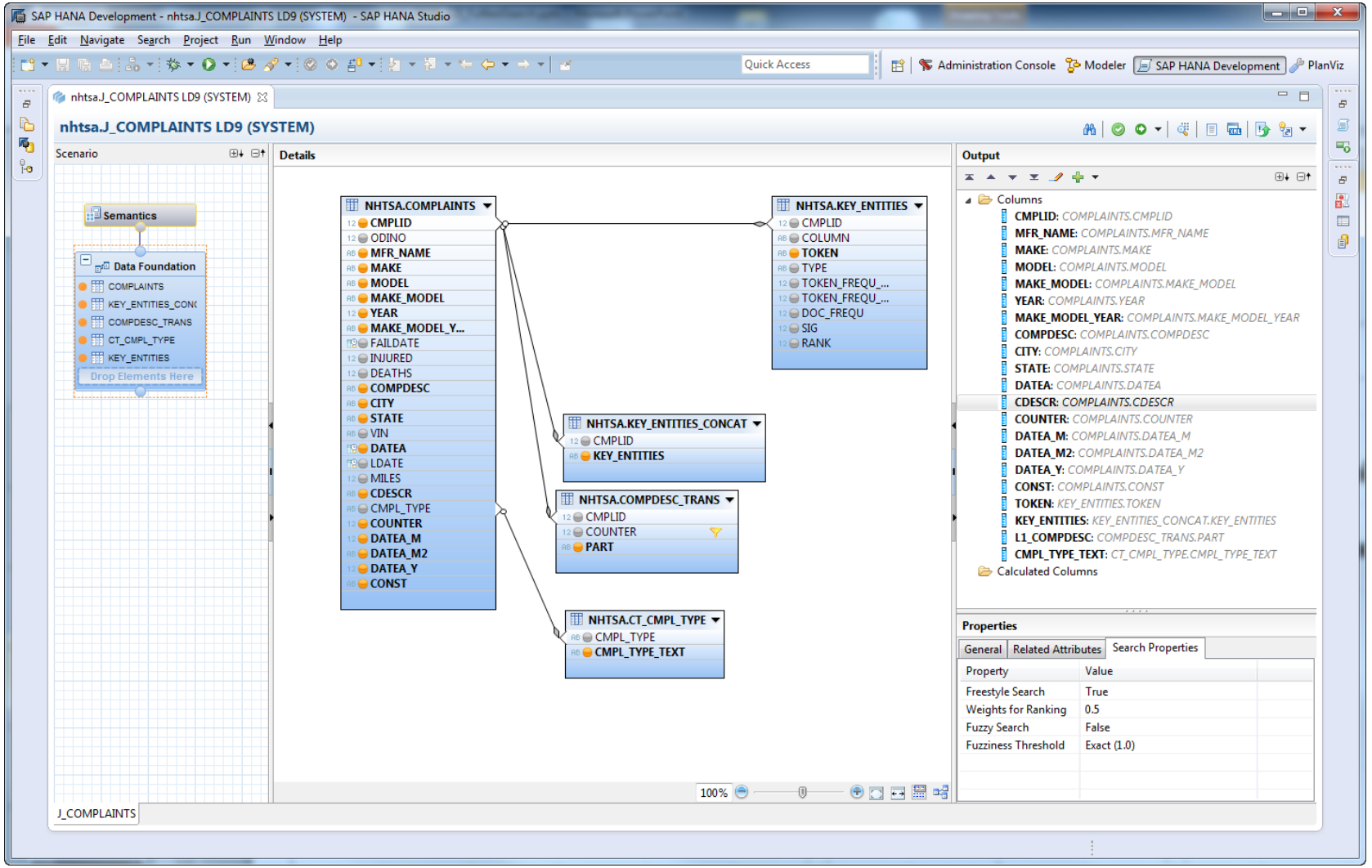
Just like analytic models, search models may comprise multiple tables

The “Search Properties” control the search behavior of your model

- Freestyle search (true/false)
- Weights for ranking ([0..1])

SAP HANA

Search Model Example



Fuzzy Search and Search Rule Sets

Fuzzy Search and Search Rule Sets

Fuzzy Search

- Find database content similar to the search terms or with typo errors
- Incomplete search terms
- Search with synonyms (term mapping)
- Duplicate prevention

SQL Features

- Fuzzy Search Index for VARCHAR columns and TEXT columns

Search Rules

- Rule Sets: multiple SELECTs in one call
- Graphical Editor
- Repository Objects
- Easy to use: Built-in function

Content specific search

- Postcode
- House number

Fuzzy search

SELECT * FROM <tablename>

WHERE CONTAINS (<column_name>, <search_string>, FUZZY (0.8))

Example:

SQL Result

```
select CID,CONTENT from COMMENTS
WHERE CONTAINS(CONTENT,'javascript',FUZZY(0.6))
```

	CID	CONTENT
1	1,816	构建iPhone企业级应用：基于HTML CSS 和JavaScript构建iPhone企业级应用：基于HTML CSS
2	7,496	javaTCP/IP方面的好书，帮助我学习javaTCP/IP编程的主要过程
3	10,758	JavaScript基础教程（第7版）循序渐进地讲述了JavaScript 及相关的CSS 、DOM 与Ajax 等...
4	10,791	对c#,ASP.NET熟悉些,想学点javascript 翻了一下,内容比较多,对例子的解释不够详细,可能是...
5	15,448	非常好的Javascript书籍 强烈推荐学习web前端的同学好好读一下！

SimilarCalculationMode

SimilarCalculationMode to control the similarity algorithm

Mode	Impact on wrong characters	Impact on additional characters in search	Impact on additional characters in table
search	high	high	low
compare (default)	moderate	high	high
symmetricsearch	high	moderate	moderate
substringsearch	high	high	low

Search Rule Sets

Example: search for all persons similar to a given set of data

- Find all records that have
  - the same name + address OR
  - the same name + date of birth OR
  - the same last name + address (find persons in same household)
- Use case: For example, search for possible duplicates before saving a new record

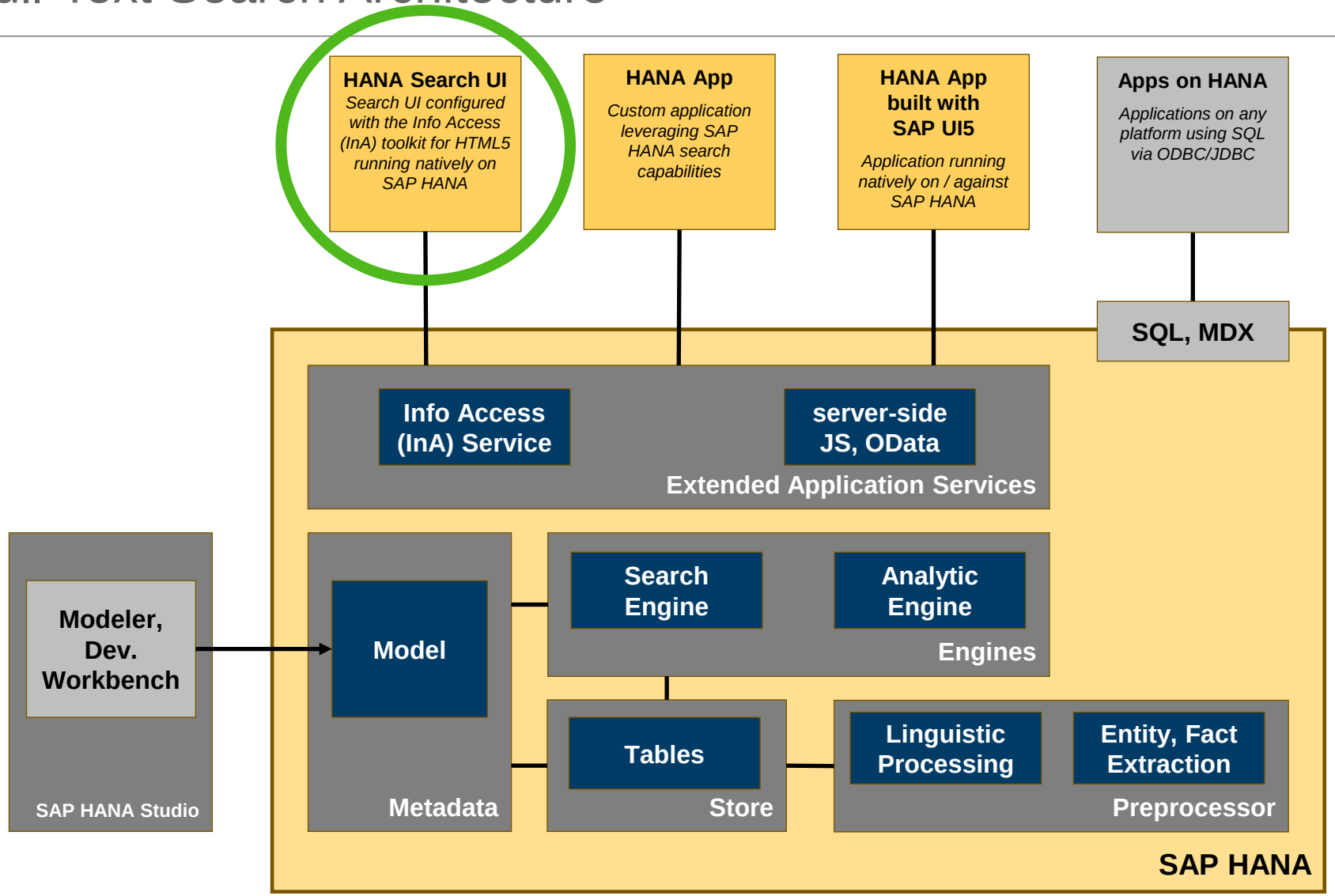
Search Rules

- define criteria for two records to be considered similar
- define which records are returned
- multiple sets of rules can be defined for different use cases

User Interface, SAP HANA Info Access (InA) toolkit

SAP HANA

Full-Text Search Architecture



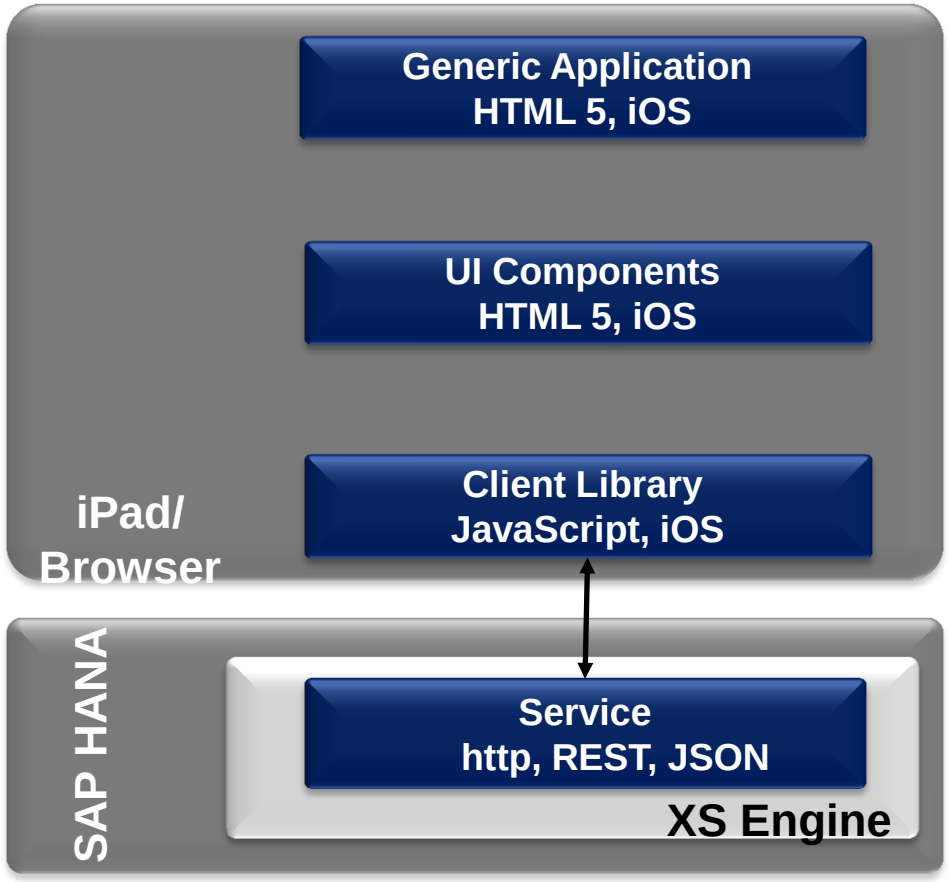
SAP HANA Info Access (InA) toolkit,
SInA API for JavaScript

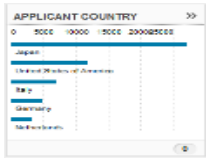
Full-Text search via SQL

Modeling,
Search Rule Sets

Full-Text Indexing, Text Analysis

Introducing the InA Family






```
req.onreadystatechange=function(){
facet =  response.getFacet('sales');
 for (item in facet){
   segment.title = item.title;
   segment.value = item.value;
   pieChart.addSegment(segment);
 }
}
{
 "d": {
  "AnalyticResult": { … },
  "Suggestions": {
      "Inlinecount": {
          "__deferred": { … }
       },
      "Items": [
          {
           "Attribute": "PRODUCT_NAME",
           "Score": "67414",
           "Term": "ANDRE BRUT 750ML"
          },
…
}
```

HANA InA App for iOS
HANA InA App for HTML

HANA InA Toolkit for iOS
HANA InA Toolkit for HTML

Information Access (InA) Service

SAP HANA Info Access (InA) toolkit for HTML5

Toolkit to configure modern, highly interactive search UIs

Shipped with HANA and included in the HANA license

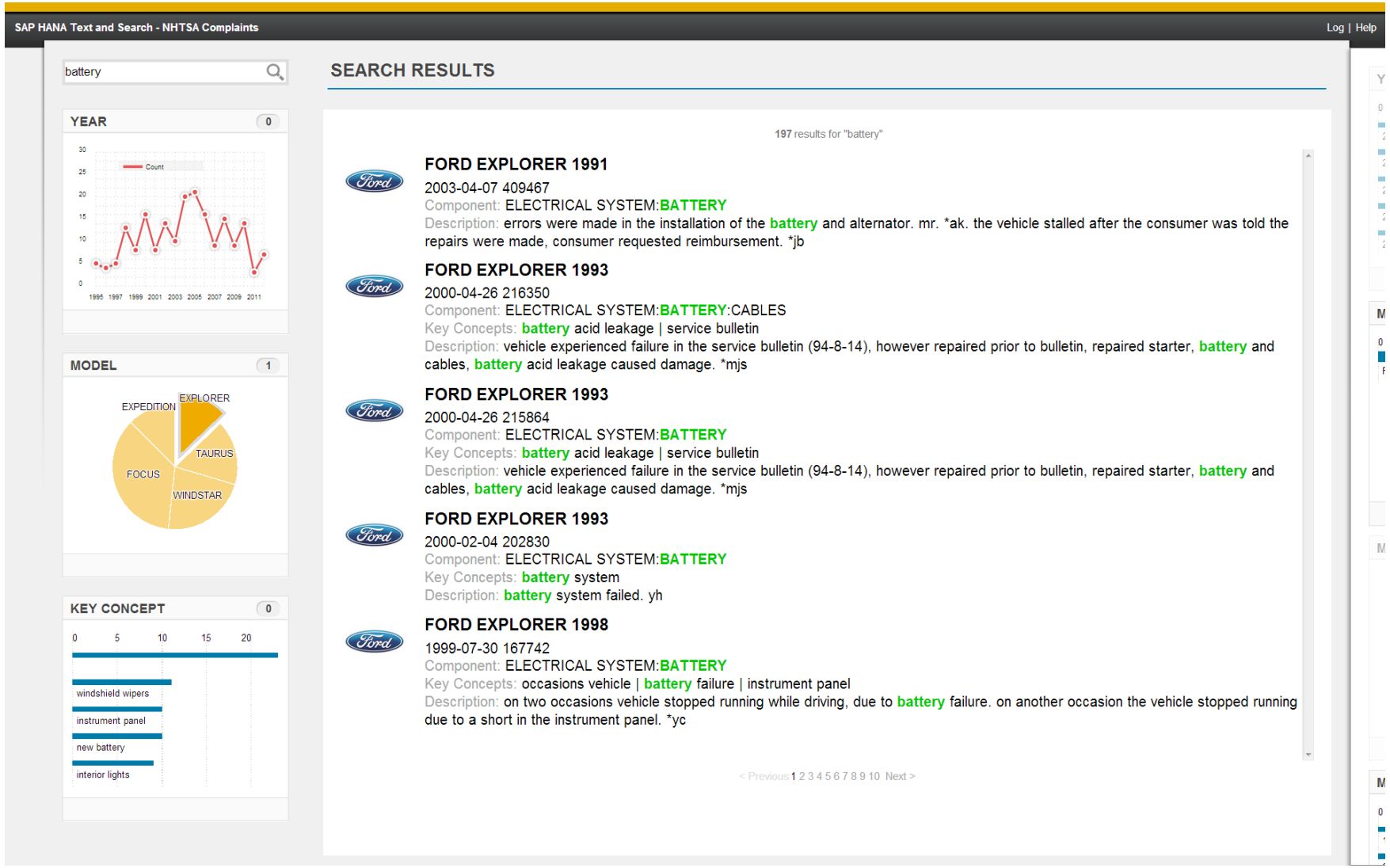
To use the toolkit you..

- import the InA toolkit Delivery Unit with SAP HANA Studio
- work with the SAP HANA Studio development tools to
  - copy existing HTML template
  - enter name of your search model in the template
  - configure which attributes are exposed as facets
  - configure layout of results list and detail screen

The InA toolkit is NOT a general purpose UI framework

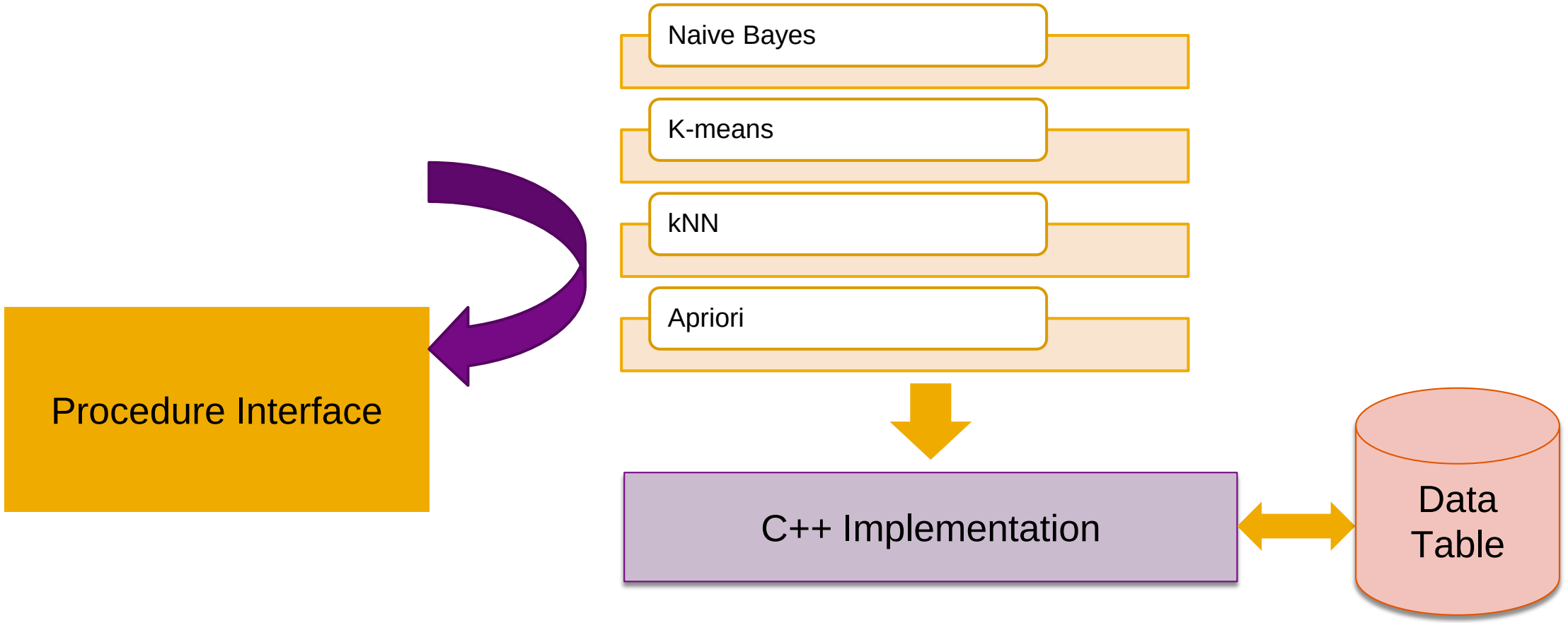
SAP HANA Info Access (InA) toolkit for HTML5

Example UI



Text Analysis with HANA Predictive Analysis Libraries and R

SAP HANA Predictive Analysis Library



Use PAL

1. Table, data type preparation

2. Generate store procedure:

```
CALL SYSTEM.AFL_WRAPPER_GENERATOR('<procedure_name>', '<area_name>',
'<function_name>', <signature_table>);
```

3. Call algorithm

```
CALL <procedure_name>(<data_input_table> {,…}, <parameter_table>, <output_table> {,
…}) with overview;
```

PAL Example – K-Means

1. Prepare the Data

```
CREATE COLUMN TABLE "TELCO" (

"ID" INTEGER NOT NULL, "AVG_CALL_DURATION" DOUBLE,  "AVG_NUMBER_CALLS_RCV_DAY" DOUBLE, "AVG_NUMBER_CALLS_ORI_DAY" DOUBLE, -"DAY_TIME_CALLS" DOUBLE”

"WEEK_DAY_CALLS" DOUBLE,  "CALLS_TO_MOBILE" DOUBLE,  "SMS_RCV_DAY" DOUBLE,  "SMS_ORI_DAY" DOUBLE,  PRIMARY KEY ("ID"))
```

2. Generate the PAL procedure

2.1 Generate table type

```
CREATE TYPE PAL_KMEANS_RESASSIGN_TELCO AS TABLE("ID" INT, "CENTER_ASSIGN" INT, "DISTANCE" DOUBLE); // Output parameter

CREATE TYPE PAL_KMEANS_CENTERS_TELCO AS TABLE("CENTER_ID" INT,"V000" DOUBLE,"V001" DOUBLE,"V002" DOUBLE,"V003" DOUBLE, "V004" DOUBLE,"V005" DOUBLE,"V006" DOUBLE,"V007" DOUBLE);  //Output parameter

CREATE TYPE PAL_KMEANS_DATA_TELCO AS TABLE("ID" INT,"AVG_CALL_DURATION" DOUBLE,"AVG_NUMBER_CALLS_RCV_DAY" DOUBLE, "AVG_NUMBER_CALLS_ORI_DAY" DOUBLE,"DAY_TIME_CALLS" DOUBLE,"WEEK_DAY_CALLS" DOUBLE,"CALLS_TO_MOBILE" DOUBLE, "SMS_RCV_DAY" DOUBLE, "SMS_ORI_DAY" DOUBLE,primary key("ID"));   //Input parameter

CREATE TYPE PAL_CONTROL_TELCO AS TABLE("NAME" VARCHAR (50),"INTARGS" INTEGER,"DOUBLEARGS" DOUBLE,"STRINGARGS" VARCHAR (100));  //specify  the different parameters to run the KMeans Algorithm
```

PAL Example – K-Means cont.

```
CREATE COLUMN TABLE PDATA_TELCO("ID" INT,"TYPENAME" VARCHAR(100),"DIRECTION" VARCHAR(100) );
```

2.2 Fill the table

```
INSERT INTO PDATA_TELCO VALUES (1, '_SYS_AFL.PAL_KMEANS_DATA_TELCO', 'in');

INSERT INTO PDATA_TELCO VALUES (2, '_SYS_AFL.PAL_CONTROL_TELCO', 'in');

INSERT INTO PDATA_TELCO VALUES (3, '_SYS_AFL.PAL_KMEANS_RESASSIGN_TELCO', 'out');

INSERT INTO PDATA_TELCO VALUES (4, '_SYS_AFL.PAL_KMEANS_CENTERS_TELCO', 'out');
```

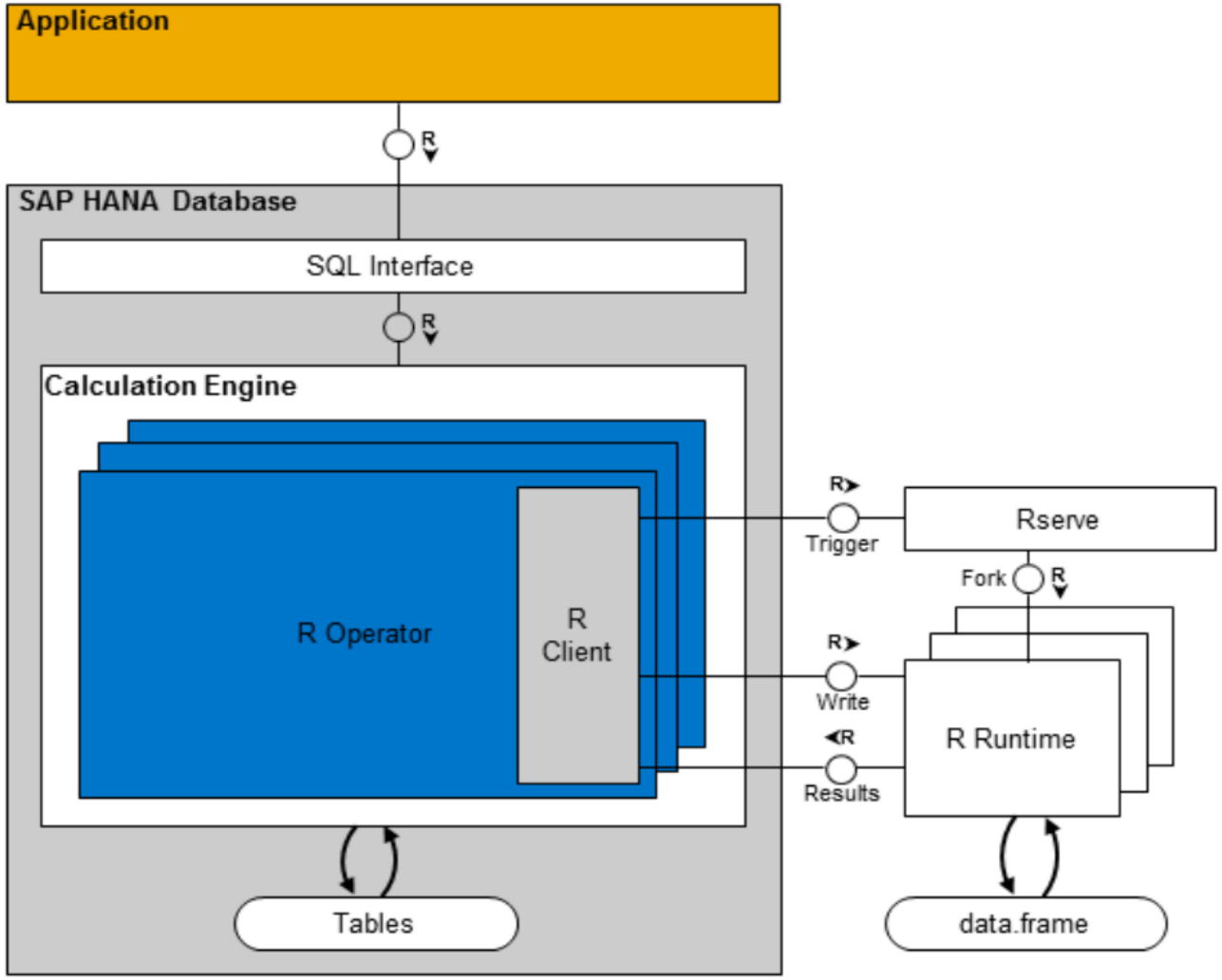
2.3 Creates the KMeans procedure that executes the KMeans Algorithm

```
call SYSTEM.afl_wrapper_generator('PAL_KMEANS_TELCO', 'AFLPAL', 'KMEANS', PDATA_TELCO)
```

3. Fill the data and run store procedure

```
CALL PAL_KMEANS_TELCO(TELCO, PAL_CONTROL_TAB_TELCO, PAL_KMEANS_RESASSIGN_TAB_TELCO, PAL_KMEANS_CENTERS_TAB_TELCO) with overview;
```

SAP HANA + R



Thank you!