



# SAP Esri Spatial Hackathon

**Palm Springs, California, March 3rd - 5th, 2018**



Hinnerk Gildhoff / January 16, 2018

# 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.

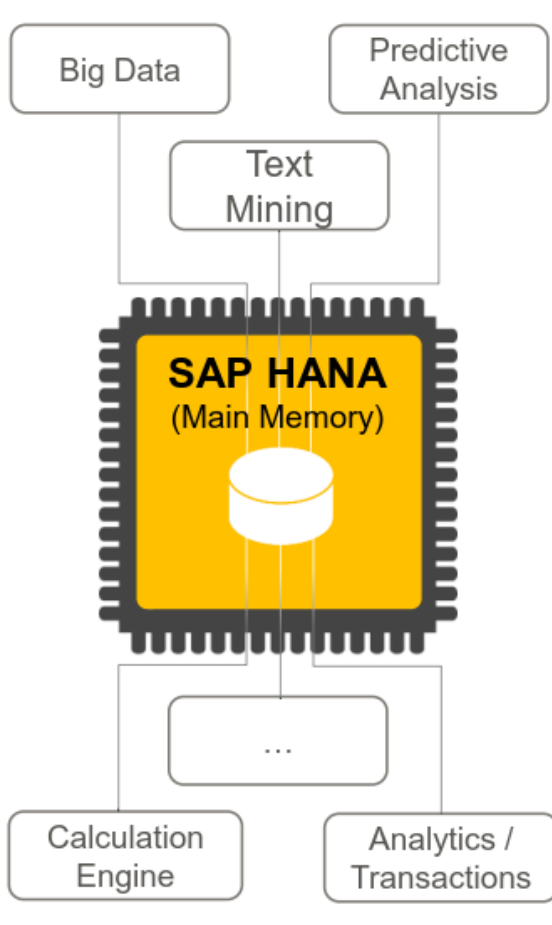
# Agenda

- SAP HANA
- HANA Spatial
- Esri Integration



# SAP HANA

A common Database Approach for OLTP and OLAP Using an In-Memory Column Database



**100%** **In-Memory** computing  
OLTP & OLAP in real-time



**No**

**Aggregates**  
On-the-fly data models  
without duplicates



**Column** and row storage



**Less**

**Indices**  
Flexible and fast  
retrieval of the dataset



**5-30x** **Compression**  
Based on column storage



**Less**

**Code lines**  
Less complexity in data  
models and code



**10-10,000x** **Acceleration**  
Massive parallelization



**Parti-  
tioning**

**Mass data analysis**

# SAP HANA

## Basic Technology - In-Memory Computing

### Keep Data in memory to speed up data access

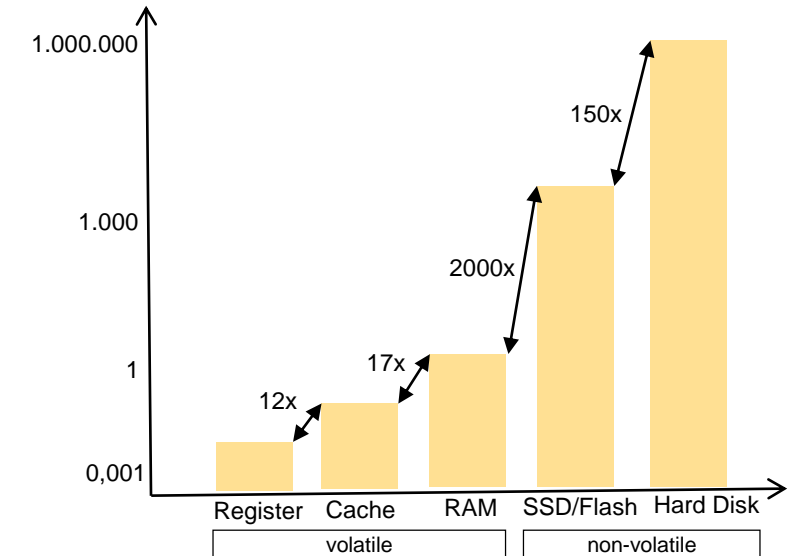
- Enterprise server with terabytes of main memory (scale-up)

### Minimize data movement by

- column store
- compression
- performing calculations at database level

### Divide and Conquer

- Use multi-core architecture of processors, multi-processor servers
- scale-out into a distributed landscape



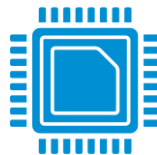
1	123	12	2	200	32	3	333	18
---	-----	----	---	-----	----	---	-----	----

1	2	3	123	200	333	12	32	18
---	---	---	-----	-----	-----	----	----	----

Id	Date	Name	#	Name
1	123	12	12	John
2	200	32	32	Flore
3	333	18	18	Lee



IN-MEMORY,  
ACID-COMPLIANT,  
COLUMN BASED



MULTI-CORE/  
PARALLELIZATION



ADVANCED  
COMPRESSION

# SAP HANA Architecture

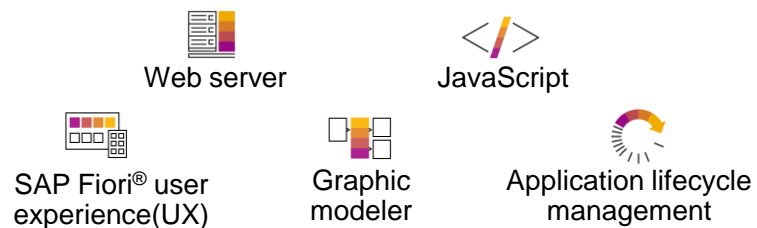
— All Devices —

— SAP, ISV and Custom Applications —

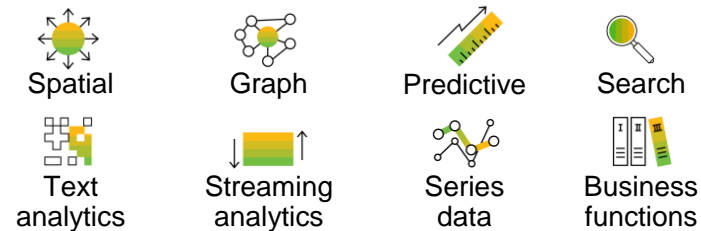
## SAP HANA® Platform

On premise | Cloud

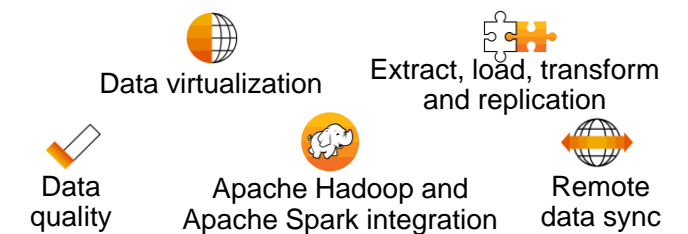
### Application development



### Advanced analytical processing



### Data integration and quality



### Database management



**ONE Open Platform**

**OLTP + OLAP**

**ONE Copy of the Data**

# Geospatial Technologies

- Rapid development of geospatial technologies in recent years
- Enables new ways of collection and analysis of geo- and business data
  - Remote Sensing (RS)
  - Unmanned Aerial Vehicle (UAV)
  - Sensors
  - Global Positioning System (GPS)
  - Geo Enrichment
  - Geographic Information System (GIS)
  - Information technologies (database)





# SAP HANA Spatial

Reveal a new dimension of your business data

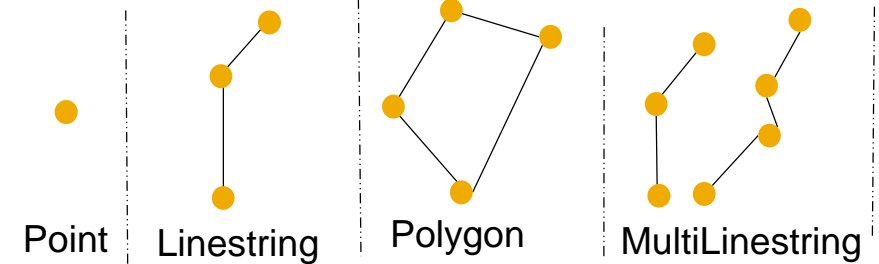
## HANA Spatial Engine

- OGC compliant
- Geodatabase for GIS
- Advanced geo-analytics
- Performance & Compression

## HANA Spatial & Graph Services

- Earth Observation, UAV and GPS Service
- Based on open and commercial geo-referenced data
- Open & Extendable Ecosystem
- Exploring the world of spatial and connected data
- Spatial micro-services architecture

### Vector Data



### Raster Data



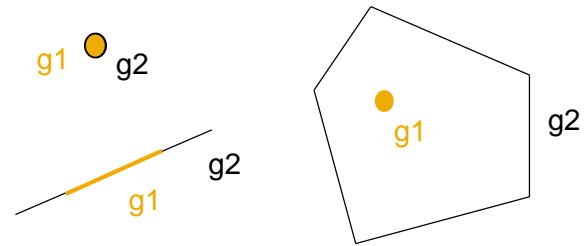
### Drone Data





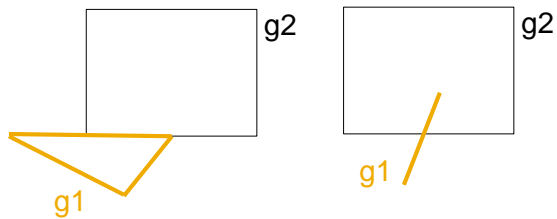
# Spatial Predicates

## SQL/MM



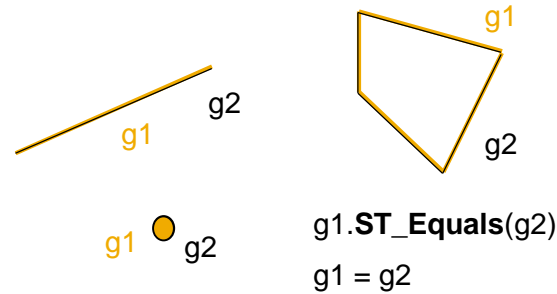
$g1.ST\_Within(g2)$

$$g1 \cap g2 = g1 \wedge I(g1) \cap E(g2) = \emptyset$$



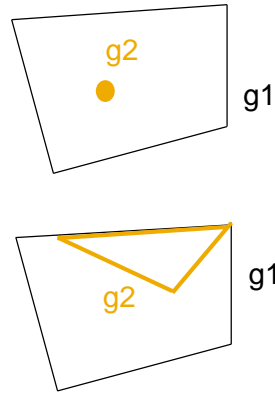
$g1.ST\_Intersects(g2)$

$$g1 \cap g2 \neq \emptyset$$



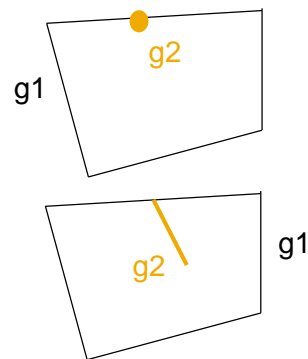
$g1.ST\_Equals(g2)$

$$g1 = g2$$



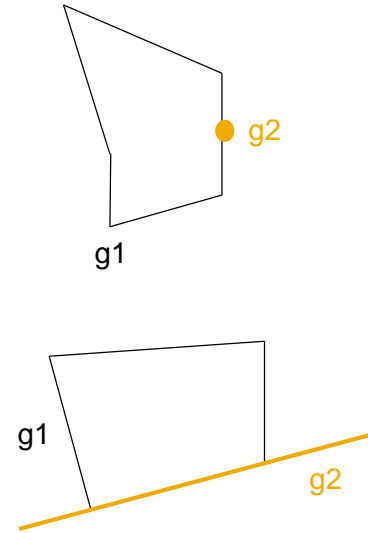
$g1.ST\_Contains(g2)$

$$g1 \cap g2 = g2 \wedge I(g1) \cap I(g2) \neq \emptyset$$



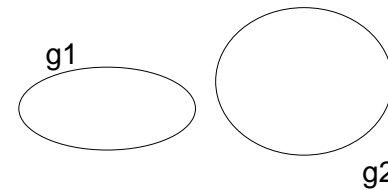
$g1.ST\_Covers(g2)^*$

$$g1 \cap g2 = g2$$



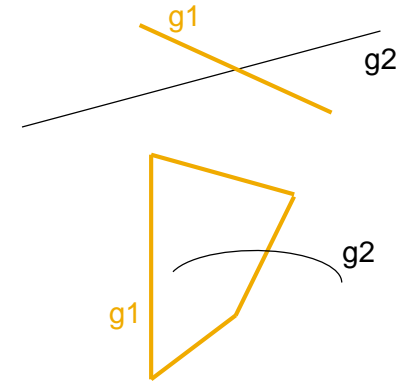
$g1.ST\_Touches(g2)$

$$(g1 \cap g2 \neq \emptyset) \wedge (B(g1) \cap B(g2) = \emptyset)$$



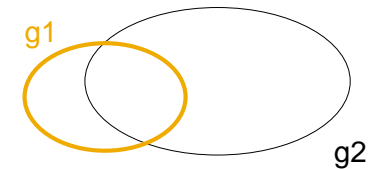
$g1.ST\_Disjoint(g2)$

$$g1 \cap g2 = \emptyset$$



$g1.ST\_Crosses(g2)$

$$I(g1) \cap I(g2) \neq \emptyset \wedge (g1 \cap g2 \neq g1) \wedge (g1 \cap g2 \neq g2)$$




$g1.ST\_Overlaps(g2)$

$$\begin{aligned} & (I(g1) \cap I(g2) \neq \emptyset) \wedge \\ & (I(g1) \cap E(g2) \neq \emptyset) \wedge \\ & (E(g1) \cap I(g2) \neq \emptyset) \end{aligned}$$

\* No OGC standard


# Consuming Spatial SQL

## 1. table creation



```
CREATE COLUMN TABLE locations
(
    id INTEGER,
    description CHAR(100),
    location ST_POINT(0)
);
```

## 2. **spatial** SQL



```
SELECT
    description
FROM
    location
WHERE
location.ST_Within(new ST_POLYGON('POLYGON((0 0,1 0,1 1,0 0))')) = 1;
```

# Consuming Spatial

## Calculation Engine

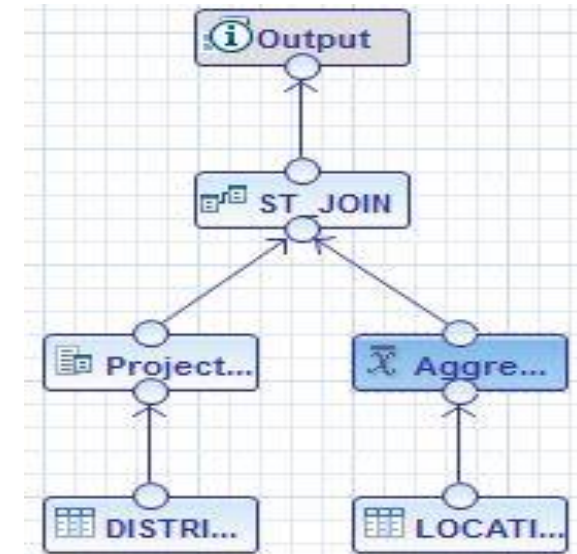
```
CREATE COLUMN TABLE locations  
(  
  id INTEGER,  
  description CHAR(100),  
  location ST_POINT(4326),  
  shape ST_Geometry(4326)  
);
```

### 1. Spatial tables

### 2. Calculation view

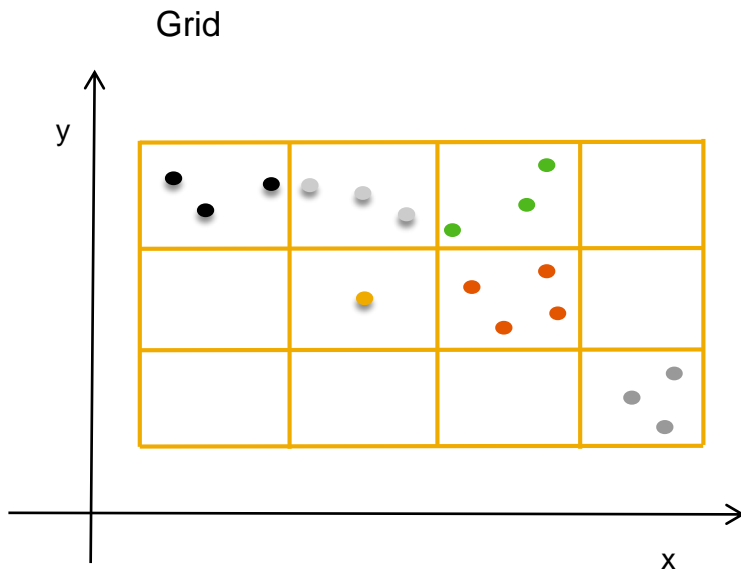
### 3. Standard SQL

```
SELECT  
  description  
FROM  
  calculation view
```

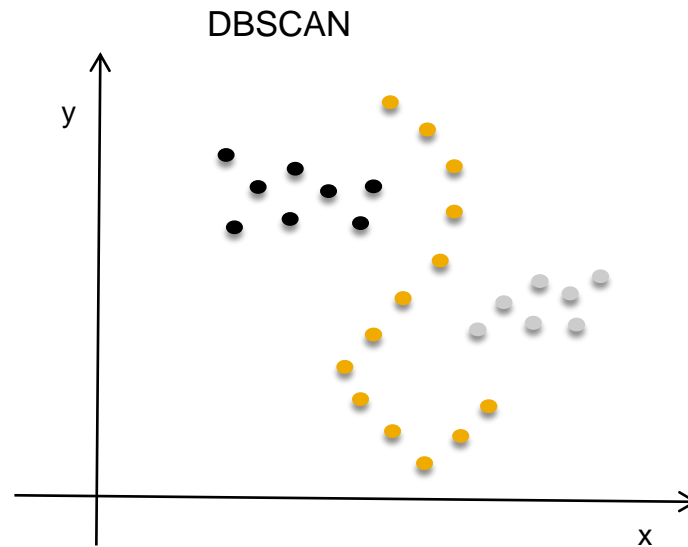


# Clustering

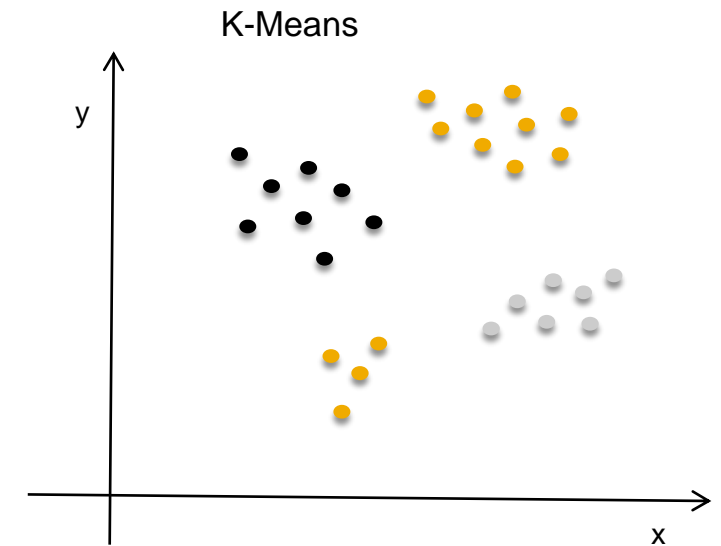
## Native Clustering Algorithms



- Good for first impression
- Easy to use
- Extreme fast grid clustering



- Best for non-spherical clusters
- Density based
- Higher complexity, better insights



- Best for spherical clusters
- Centroid based
- Higher complexity, better insights

# Clustering

## SQL Example – Use Cases

### Clustering with metadata accessor (similar to group by aggregates)

```
SELECT
  ST_ClusterId()           AS cluster_id,
  ST_ClusterCentroid()     AS centroid,
  ST_ClusterEnvelope()     AS envelope,
  count(*)                 AS num_hholds,
  avg(income)              AS avg_clus_income
FROM households
WHERE income > 120000
GROUP CLUSTER BY location USING DBSCAN EPS 4 MINPTS 1000
HAVING count(*) >= 300
```

Cluster_id	Centroid	Envelope	Num_hholds	Avg_clus_income
1	Point(5 4)	Polygon((...))	311	304.123
2	Point(15 78)	Polygon((..))	621	714.234
...				

### Identify for each point the corresponding clusters (similar to window functions)

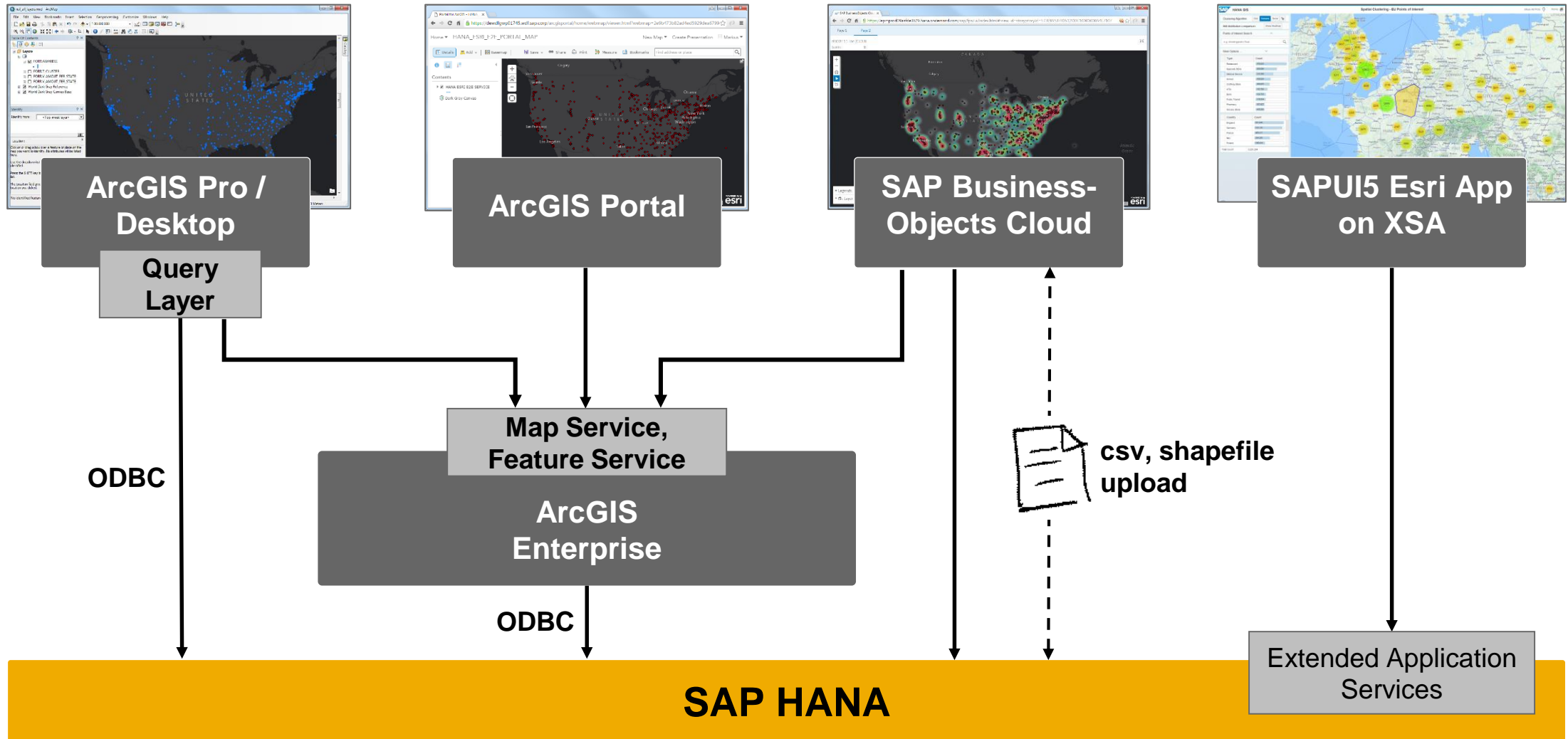
```
SELECT
  ST_ClusterId()
  OVER (CLUSTER BY location USING KMEANS CLUSTERS 5 ) AS cluster_id,
  vm_id,
  location,
  revenue
FROM vending_machines
WHERE revenue < 15000
ORDER BY cluster_id, revenue
```

Cluster_id	Vm_id	Location	Revenue
1	1	Point(1 1)	1.311
1	2	Point(1.4 1.2)	1.166
1	3	Point(1.2 1.3)	799
2	4	Point(5.3 5.0)	2.125
2	5	Point(5.7 6)	1.750
3	6	Point(20 20)	1.532
...			

# SAP HANA Spatial Esri Applications

ArcGIS Online

SAP Cloud Platform





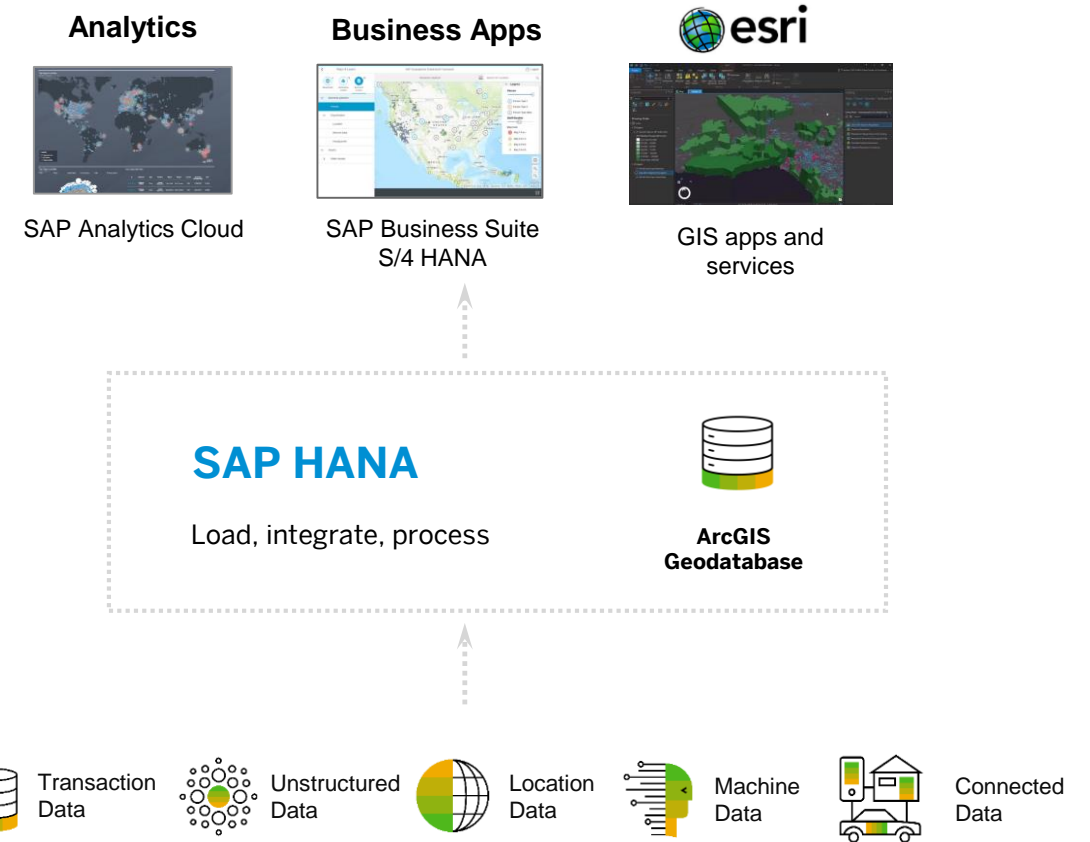
# Geodatabases Support for HANA

ArcGIS 10.6 and ArcGIS Pro 2.1

## Geodatabase Support for HANA

- Subtypes
- Domains
- Relationship classes
- Attachments
- Editor tracking
- Non-versioned archiving
- Offline editing with sync capabilities
- New service based transaction model - long transactions
- Utility network

Requires SAP HANA 2.0 SPS 2





A satellite with large solar panels is shown in orbit above the Earth's surface. The satellite is gold-colored with multiple black solar panel arrays extending from its body. It is pointing towards the Earth, and a green beam of light is visible emanating from its sensor. The Earth's blue oceans and white clouds are visible below, with the blackness of space and the curve of the horizon in the background.

#SAPEsri

#DevSummit

#SAPHANA

#ESRI

**Happy Mapping!**

[SAP HANA Spatial Reference  
Earth Observation Service](#)



[SAP HANA Academy  
SAP HANA & Esri](#)

**Thank You !**

[hinnerk.gildhoff@sap.com](mailto:hinnerk.gildhoff@sap.com)  
[@HinnerkGildhoff](https://twitter.com/HinnerkGildhoff)

