

Demo:

Vector Search in InterSystems IRIS

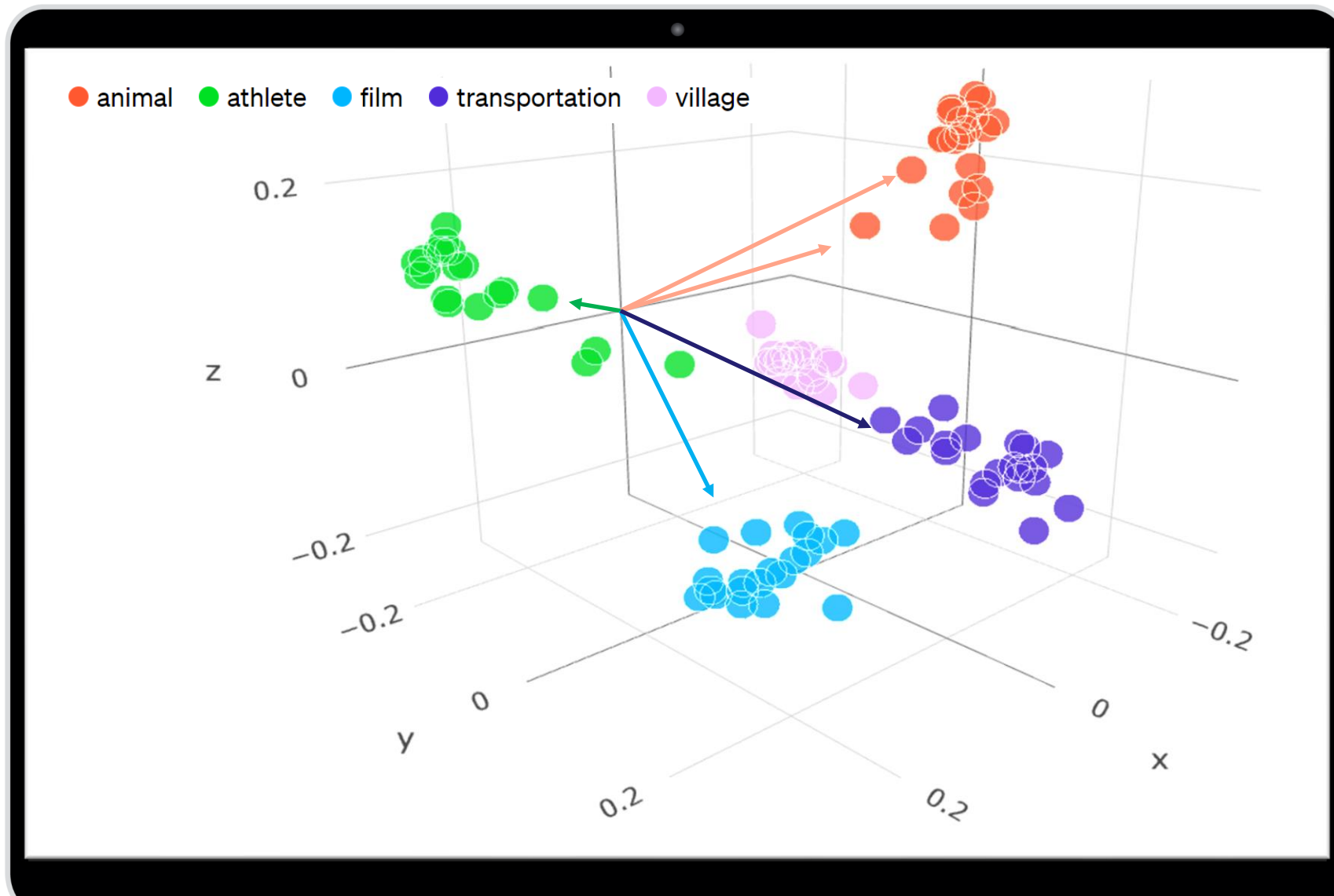


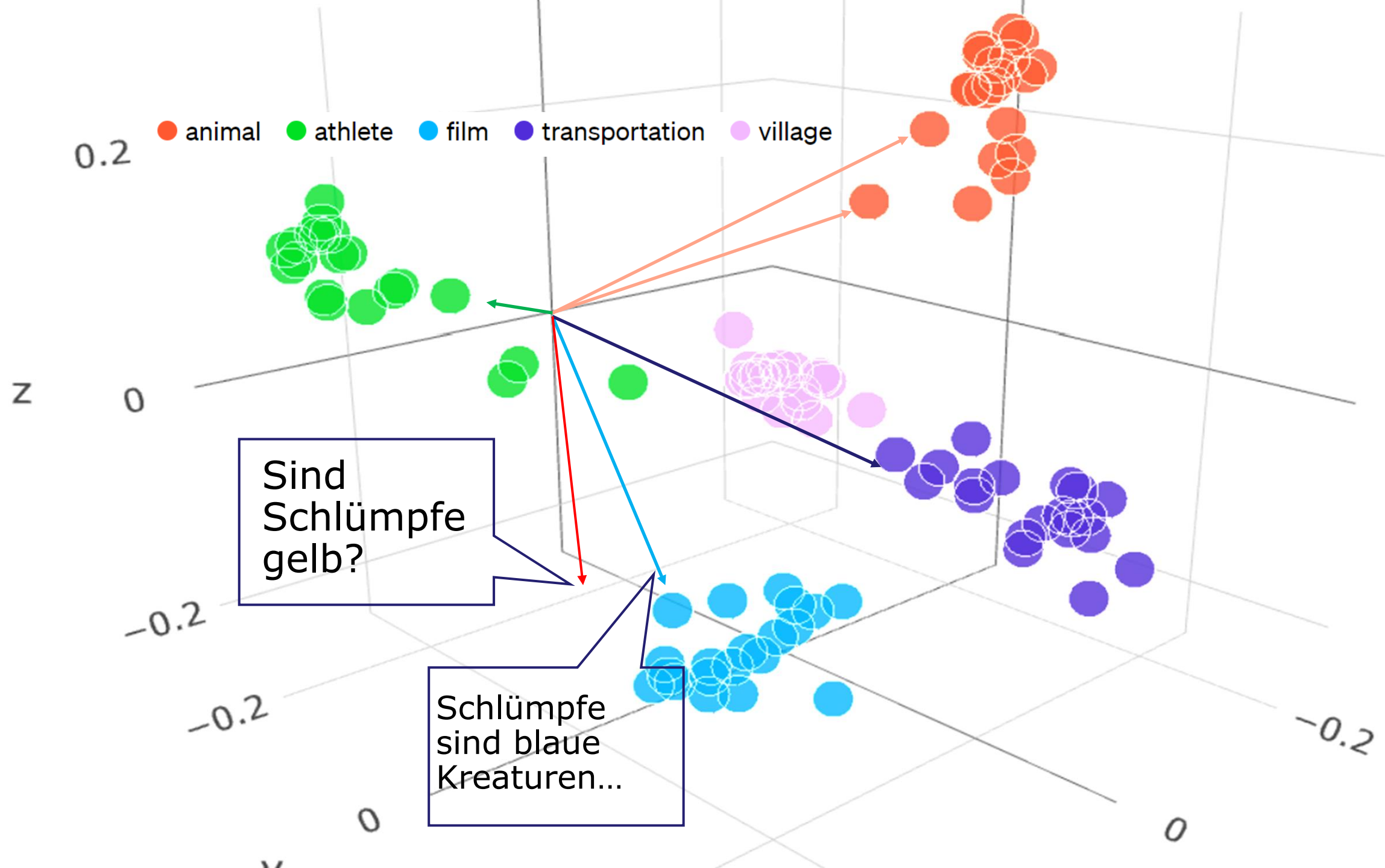
Michael Braam
Senior Sales Engineer



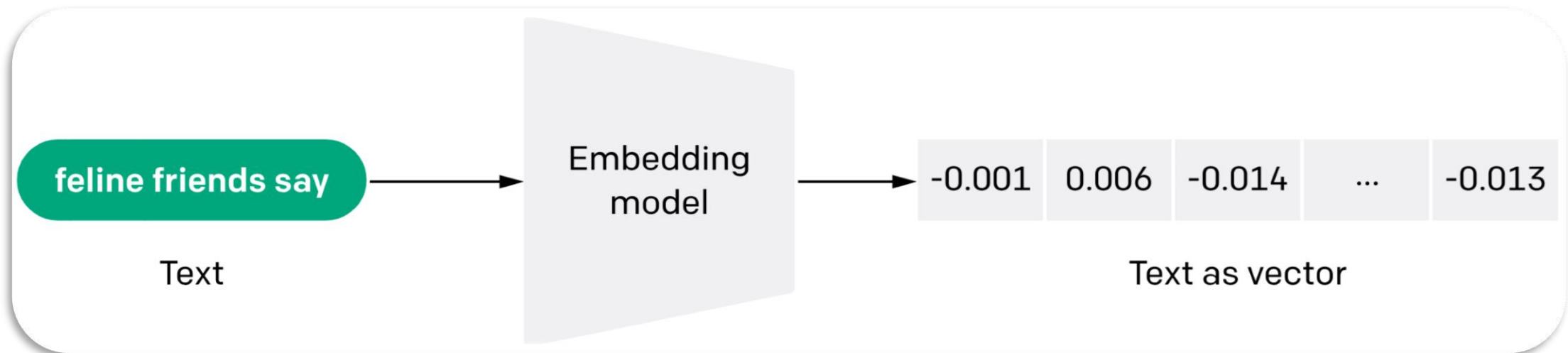
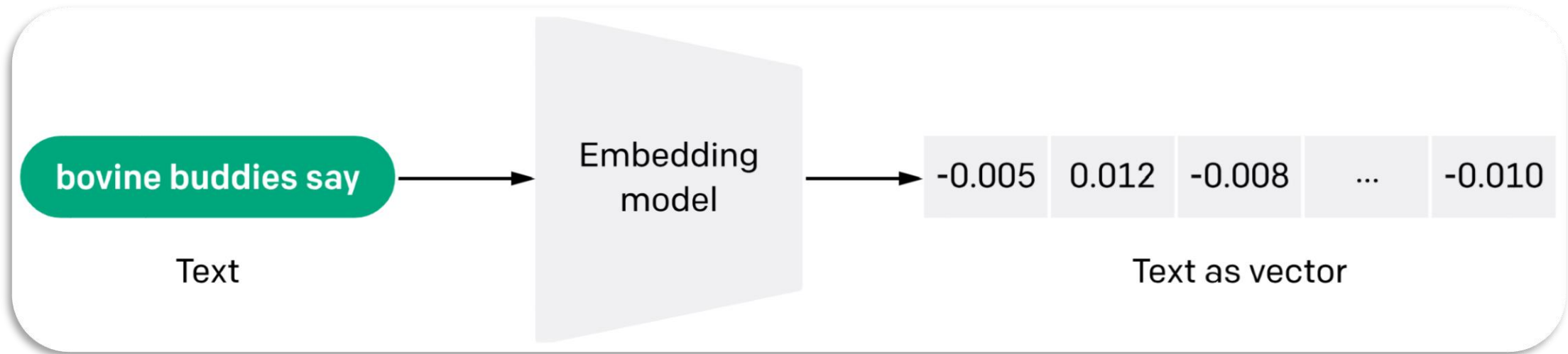
Felix Vetter
Sales Engineer Intern

Sätze & deren Kontext als Vektoren





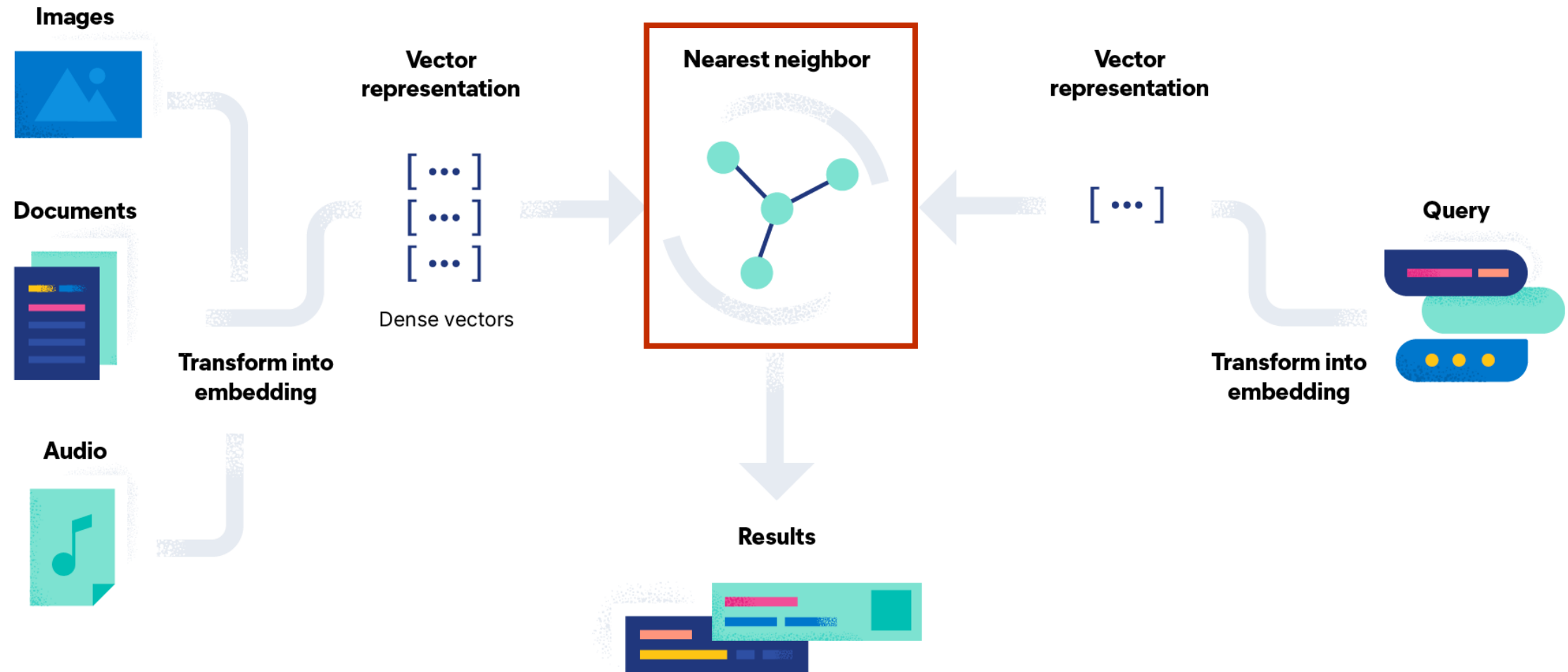
Word Embeddings



Vector Search



- **Vector search** stellt Datenelemente durch numerische Vektoren dar, was die Indizierung und Ausführung von Abfragen über numerische Inhalte ermöglicht



InterSystems IRIS Vector Search (aktuell)



- VECTOR Datentyp & Ähnlichkeits Berechnungen
 - CREATE TABLE t (txt VARCHAR(1000), vec VECTOR(INT, 200));
 - INSERT INTO t VALUES ('...', TO_VECTOR('1,2,3,...', INT));
 - SELECT TOP 10 * FROM t
ORDER BY VECTOR_COSINE(vec, TO_VECTOR('1,3,4,...', INT)) DESC;

InterSystems IRIS Vector Search (demnächst)



- embedding config
 - `INSERT INTO %Embedding.Config (Name, ...) VALUES ('openai1', ...)`
- neuer VECTOR Datentyp & Ähnlichkeits Berechnungen
 - `CREATE TABLE t (txt VARCHAR(1000), emb EMBEDDING('openai1', 'txt'));`
 - `INSERT INTO t VALUES ('some text...');`
 - `SELECT TOP 10 * FROM t
ORDER BY VECTOR_COSINE(EMBEDDING('some similar text...'), emb) DESC`

Verfügbar für Kunden via Vector Search Early Access program

Nearest Neighbor Search



- Ziel: die nächstgelegenen Vektoren zu einem gegebenen Abfragevektor finden
- Exakte Nearest Neighbor Suche:
 - Berechnung der Abstände zwischen dem Abfragevektor und allen Datenvektoren und Auswahl der nächstgelegenen Vektoren
 - Liefert genaue Lösungen, ist aber langsam
- Approximate Nearest Neighbor (ANN) Suche:
 - Schnelles Finden einer Näherungslösung für die Suche nach dem nächsten Nachbarn, indem eine Datenstruktur aufgebaut wird, die den Suchraum verkleinern kann

Hierarchical Navigable Small World (HNSW)



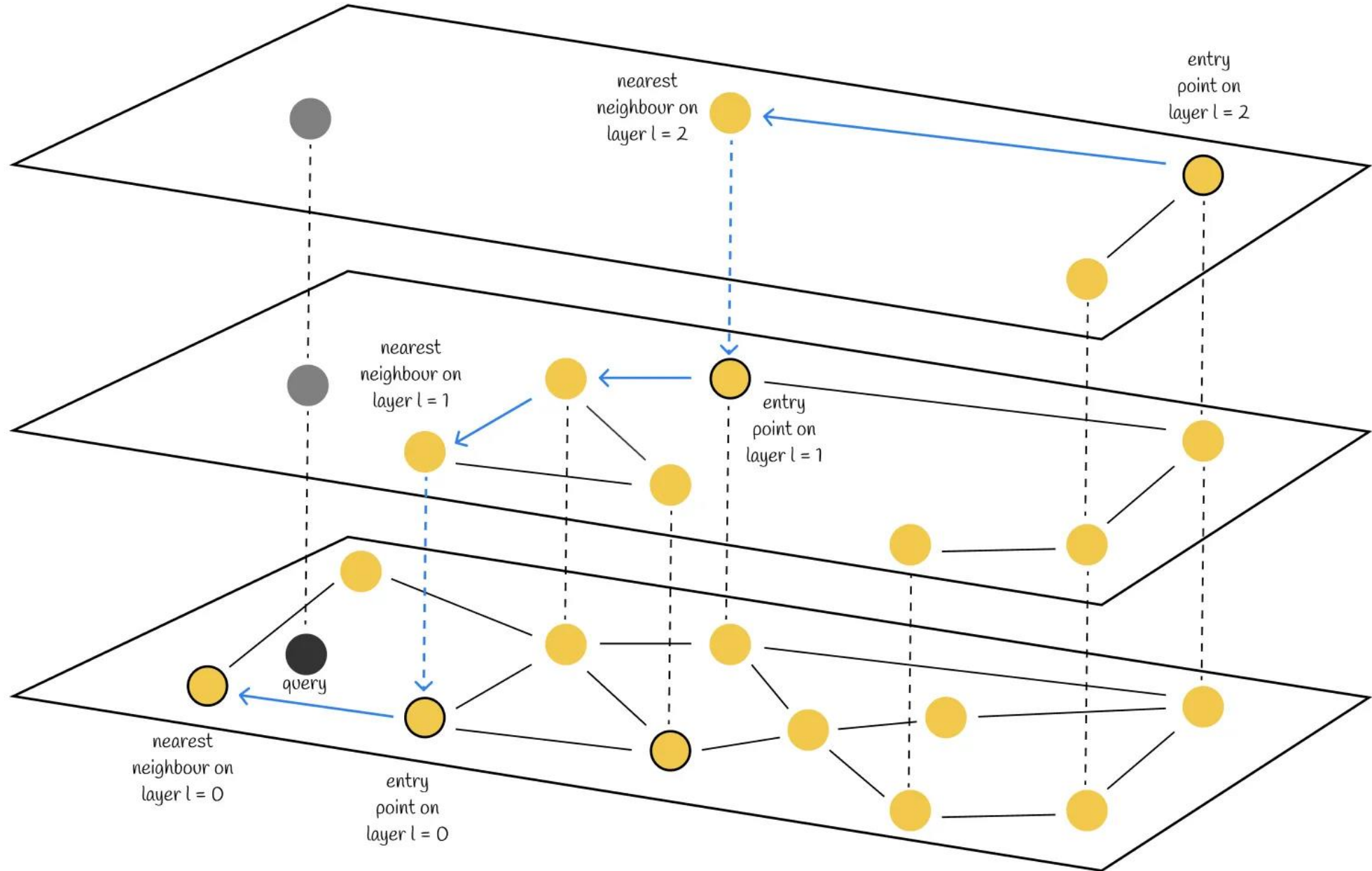
- Der HNSW Algorithmus ist eine graph basierte approximate nearest neighbor (ANN) search Technik, welche in vielen Vektordatenbanken implementiert ist
- Die generelle Idee:
 - Aus den Datenvektoren ein mehrschichtiges Graphenmodell erstellen. In jeder Schicht ist jeder Datenvektor mit anderen Datenvektoren verbunden, die in der Nähe liegen
 - Globals sind ein guter Weg solche Graphen zu speichern
 - Bei einem Abfragevektor den mehrschichtigen Graphen verwenden, um schnell eine Teilmenge der Datenvektoren zu ermitteln, die dem Abfragevektor nahe kommen, um dann nur unter diesen Punkten zu suchen.



layer 2

layer 1

layer 0



DDL for creating HNSW



```
CREATE INDEX embIDX ON MyTab(vec) AS HNSW WITH DISTANCE = 'COSINE';
```

Verfügbar für Kunden via Vector Search Early Access program

Stats for 10 runs with 10 random vectors	SELECT TOP 10 id, VECTOR_COSINE(vec1, to_vector(?, DOUBLE, 32)) FROM %IGNOREINDEX * SQLUser.vtable ORDER BY 2 DESC		SELECT TOP 10 id, VECTOR_COSINE(vec1, to_vector(?, DOUBLE, 32)) FROM SQLUser.vtable ORDER BY 2 DESC	
Average Run Time	0.19039s		0.00846s (22.5x faster!)	
Average Global Reference	20008		4358.1	
Average Accuracy	100%		93%	
Example Results	ID	Expression_2	ID	Expression_2
	42702	.71951992680902920884	42702	.7195199268090292088
	5244	.71827802159096409972	5244	.7182780215909640997
	51232	.70870149768201351481	51232	.7087014976820135148
	63205	.68611315978136810933	63205	.6861131597813681093
	60709	.68294108658930408051	42331	.6601174180422056947
	42331	.66011741804220569473	69657	.652699208325380642
	69657	.65269920832538064203	86868	.6448593780305255097
	86868	.64485937803052550966	59058	.639482152861199582
	59058	.63948215286119958201	89998	.6384857777267687329
	89998	.63848577772676873287	15033	.6342528103690919794



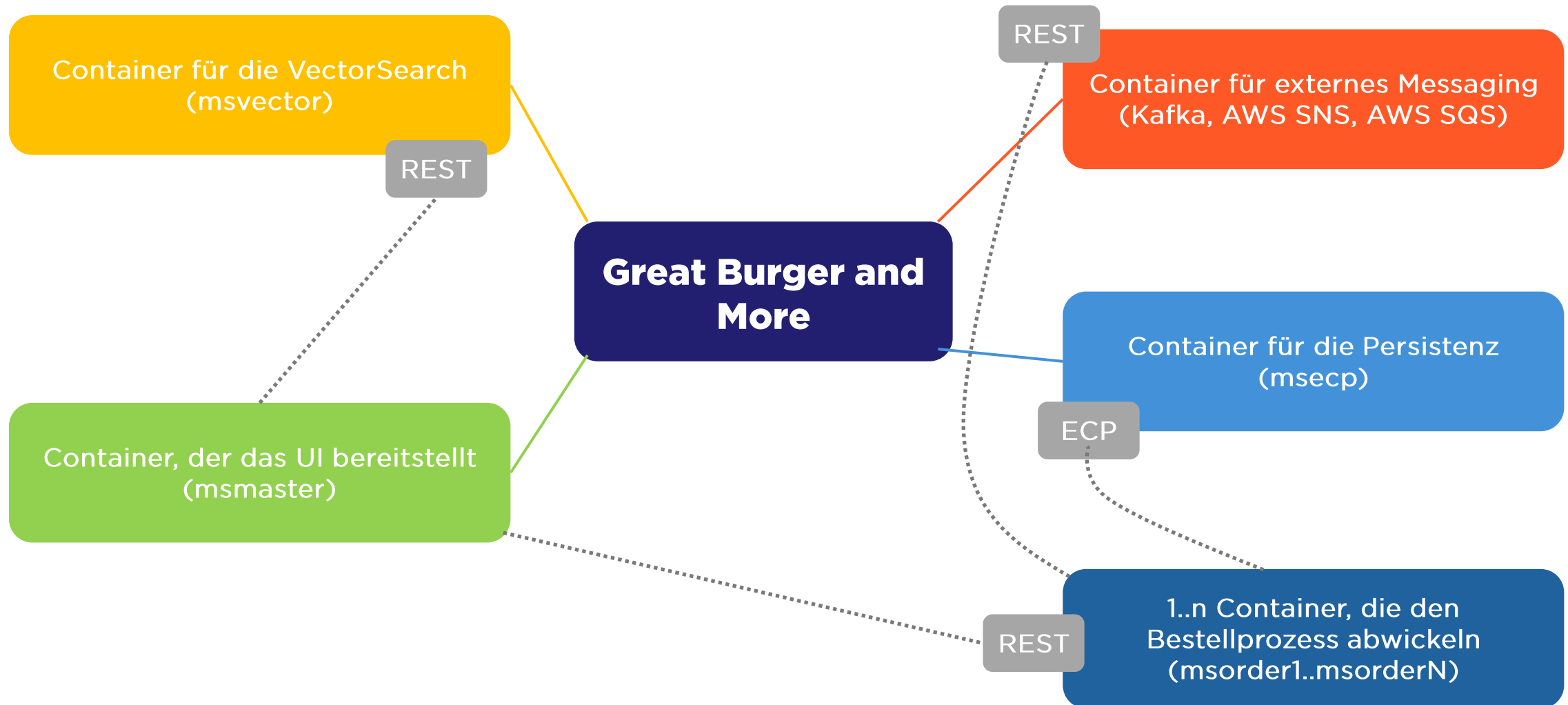
UseCases für VectorSearch

Welche Erfahrungen & UseCases habt ihr?

Demo



Legacy Anwendungen mit VectorSearch





Thank You