

# dkd

Olivier Dobberkau, 18. March 2026

## PIMP MY SOLR SEARCH



# Chatting with new Solr vector database

Olivier Dobberkau



- CEO dkd Internet Service GmbH
- President TYPO3 Association
- [olivier.dobberkau@dkd.de](mailto:olivier.dobberkau@dkd.de)
- <https://www.linkedin.com/in/olivierdobberkau/>

# The Problem with Traditional Search

---

Human searches for: “hydraulic power units”

Traditional Keyword Search:

- ✘ Misses “fluid power systems”
- ✘ Misses “pressure control valves”
- ✘ Misses “hydraulic actuators”

**Result:** Human finds nothing, leaves frustrated

# What Humans Really Want

---

» I need something that controls pressure in my industrial system«

→ **They don't care about exact keywords.**

→ **They care about finding the RIGHT solution.**

# Enter: Vector Search

---

## From Keywords to Meaning

**Traditional:** Matches words | **Vector Search:** Understands concepts



**“hydraulic power units” and “fluid power systems”**

- Are semantically similar
- Get mapped to nearby points in vector space
- Are found together!

# What Are Vectors Anyway?

---

Simple Answer: Numbers that Represent Meaning

"hydraulic power units" → [0.23, -0.45, 0.67, ...]

"fluid power systems" → [0.21, -0.43, 0.69, ...]

"apple fruit" → [-0.82, 0.34, -0.12, 0.91, ...]

Similar meanings = Similar numbers = Found together

# The Magic of Semantic Similarity

---

Example: Word Relationships

**King - Man + Woman  $\approx$  Queen**

**Paris - France + Germany  $\approx$  Berlin**

This same idea powers vector search!

# Apache Solr 9.9+ Makes this Easy

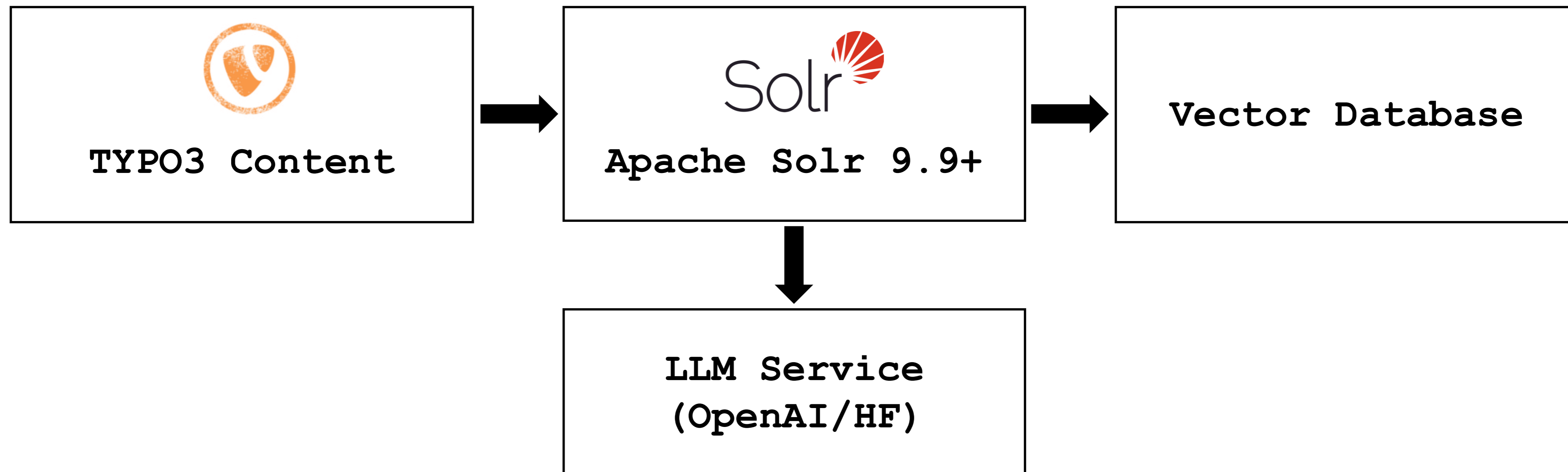
---

## Native Vector Database Support

- ✓ Built-in vector indexing
- ✓ Dense vector search (KNN)
- ✓ Text-to-Vector pipeline
- ✓ LLM integration out of the box

No need for separate vector databases!

# Architecture Overview



# Supported LLM Services

---

Choose Your Provider



**OpenAI** - GPT embeddings



**Hugging Face** - Open models



**Cohere** - Multilingual support



**MistralAI** - European alternative

All integrated via LangChain4j

# How It Works: Indexing

---

1. TYPO3 Content



2. Extract Text



3. Send to LLM → Generate Vector [0.23, -0.45, ...]



4. Store in Solr (text + vector)

**Happens automatically during indexing!**

# How It Works: Searching

---

1. Query: "pressure control for industrial systems"  
↓
2. Convert Query to Vector via Embedding Model  
↓
3. Find Similar Vectors in Solr (via KNN)  
↓
4. Return Relevant Documents

**Semantic matching, not just keywords!**



# Let's See It In Action!

---



**LIVE DEMO**

# Demo Scenario 1: Traditional Search

Query: "hydraulic power units"

Traditional Solr Response:

```
{
  "numFound": 3,
  "docs": [
    {"title": "Hydraulic Power Unit HPU-2000"},
    {"title": "Power Units for Hydraulic Systems"},
    {"title": "HPU Accessories"}
  ]
}
```

Limited to exact keyword matches



# Demo Scenario 2: Vector Search

Same Query: "hydraulic power units"

Vector Search Response:

```
{
  "numFound": 15,
  "docs": [
    {"title": "Hydraulic Power Unit HPU-2000", "score": 0.95},
    {"title": "Fluid Power Systems Overview", "score": 0.89},
    {"title": "Pressure Control Valves", "score": 0.87},
    {"title": "Hydraulic Actuators and Cylinders", "score": 0.85},
    {"title": "Industrial Pump Solutions", "score": 0.82}
  ]
}
```

Semantic understanding finds related concepts!



# Conversational Search: The Next Level

---

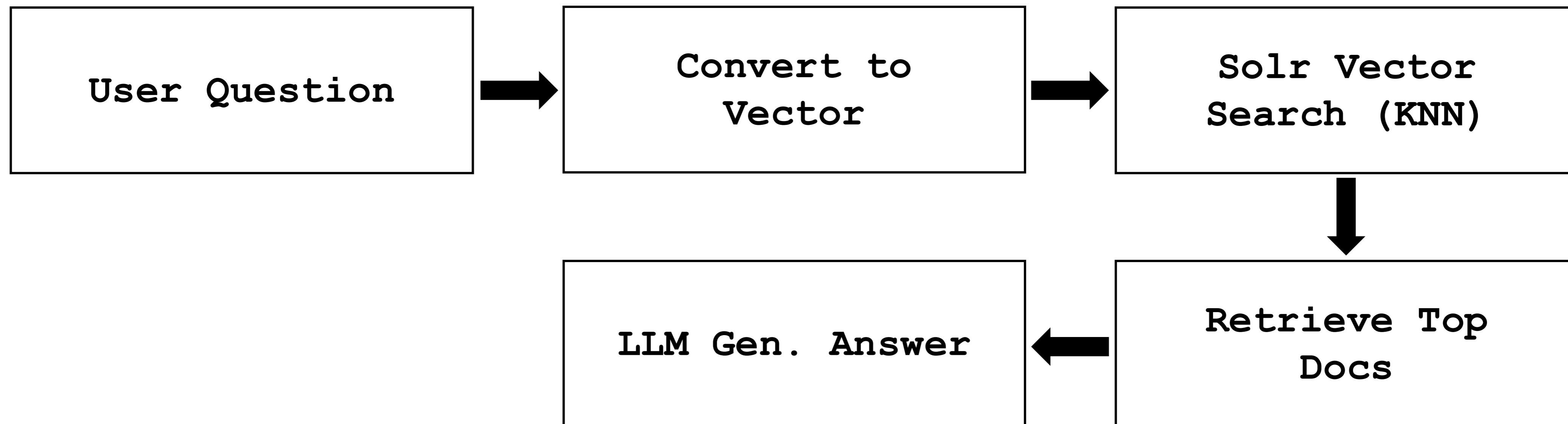
**Human asks naturally:**

“What products do you have for controlling pressure in industrial hydraulic systems?”

**Traditional Search:** Struggles with long queries

**Vector + RAG:** Understands intent, retrieves context, generates answer

# RAG Architecture with Solr



# Demo Scenario 3: Conversational Interface

dkd

## Natural Language Query:

### User:

“What products do you have for controlling pressure in industrial systems?”

### System:

1. Converts query to vector 2. Finds relevant products in Solr 3. Sends docs + question to LLM 4. Generates contextual answer

## Example Response

---

### Generated Answer:

“We offer several solutions for pressure control in industrial systems:

- 1. Hydraulic Power Unit HPU-2000** - Integrated pressure control up to 300bar
- 2. Pressure Control Valve Series PCV-400** - Adjustable relief valves
- 3. Smart Pressure Monitoring System** - Real-time pressure regulation

All products are suitable for industrial applications and meet DINstandards.”

**Source documents linked for verification**



# TYPO3 Integration

---

## EXT:solr is Your Friend

### Current State:

- ✅ Traditional search fully integrated
- 🚧 Vector search: Manual configuration
- 🌟 Future: Native vector field support

**Configuration Point:** solrconfig.xml and schema extensions

# Best Practices: Data Quality

---

Garbage In = Garbage Out



## Clean your content

- Remove boilerplate text
- Meaningful titles and descriptions
- Structured content works best



## Avoid:

- Auto-generated spam
- Duplicate content
- Empty or minimal text

# Best Practices: Chunking Strategy

Problem: Long Documents



Solution: Split into meaningful chunks

**Long Document (5000 words)**



**Split into sections**



**Chapter 1: Indexing Introduction (500 words) → Vector 1**

**Chapter 2: Schema Configuration (600 words) → Vector 2**

**Chapter 3: Query Syntax (450 words) → Vector 3**

Better retrieval precision!

# Watch Out: Cost Considerations

---

## LLM API Calls Add Up

**Example Calculation:** - 10,000 documents to index - OpenAI text-embedding-3-small: \$0.00002 per 1K tokens - Average 500tokens per document - **Cost: ~\$100 for initial indexing**

**Plus:** Query-time embedding costs

**Tip:** Consider caching frequently searched queries



# Watch Out: Latency

---

**External API = Network Dependency**

**Typical Embedding Generation:** - Local: 5-10ms - API Call: 100-300ms

**Solutions:** 1. Batch processing for indexing 2. Cache query embeddings 3. Consider self-hosted models (Hugging Face)

# Watch Out: Privacy & Compliance

---

Your Data Goes to External Services



## Consider:

- GDPR compliance
- Data processing agreements
- Sensitive content handling



## Alternative:

- Self-hosted embedding models (Hugging Face Transformers)
- Hosted Solr Plus Plan (Late April 2026)

# Performance Metrics: What to Expect

---

## Vector Search Performance

**Index Size Impact:** - 10K docs: ~2GB additional storage - 100Kdocs: ~20GB additional storage - 1M docs: ~200GB additional storage

**Query Performance:** - KNN search: 50-200ms (cached) - First-time query: +API latency (100-300ms)

# Resources

---

## Documentation & Code



### Apache Solr Docs:

<https://solr.apache.org/guide/solr/latest/query-guide/text-to-vector.html>



### Blog Post on TYP03.com:

“Apache Solr for TYP03: Vector Search Integration”



### GitHub Examples:

[github.com/apache/solr](https://github.com/apache/solr) (examples/llm/)

# TYPO3 Community Efforts

---

## What's Coming



### **EXT:solr Updates:**

- Native vector field configuration
- GUI form model management
- Pre-built RAG templates



### **Community Contributors Wanted!**

- Testing
- Documentation
- Use case sharing

# Key Takeaways

---

## What You Should Remember

- 1. Vector search is production-ready** in Apache Solr 9.8+
- 2. TYPO3 integration is straightforward** with EXT:solr
- 3. Start small, measure impact, scale up** gradually
- 4. Consider costs, latency, and privacy** upfront
- 5. Hybrid approach** (traditional + vector) often works best

# What's Next for You?

---

## Experiment!

1. **Download Solr 10** and play with examples
2. **Read the blog post** for detailed implementation
  1. <https://www.dkd.de/de/blog/apache-solr-fuer-typo3-der-weg-zur-intelligenten-suche/#c11248>
3. **Join TYPO3 Slack #solr** channel for questions
4. **Share your results** with the community


**The future of search is semantic!** 

# Questions?

**Let's Chat!**

**Olivier Dobberkau**

 [olivier.dobberkau@dkd.de](mailto:olivier.dobberkau@dkd.de)

 [linkedin.com/in/olivierdobberkau](https://www.linkedin.com/in/olivierdobberkau)

 [@dkd\\_de](https://twitter.com/dkd_de)

## **Resources:**

- Blog: [typo3.com](https://typo3.com)
- dkd: [dkd.de](https://dkd.de)
- TYPO3 Association: [typo3.org](https://typo3.org)

# Thank You!



**Now Go Build Amazing Search Experiences!**



**May your searches be semantic and your results relevant!**

**dkd**

Thanks for your Time!