Elastic Stack

Modul 01: Einführung

Über dieses Modul

Dieses Modul gibt dir einen Überblick über die Grundlagen des Elastic Stack. Du lernst, wie Suchmaschinen funktionieren, wie Daten in Elasticsearch gelangen und warum Mappings für deine Analysen wichtig sind.

Lernziele

Nach diesem Modul kannst du:

- Erklären, was eine Suchmaschine von einer Datenbank unterscheidet
- Die Komponenten des Elastic Stack benennen und ihre Rollen beschreiben
- Den Unterschied zwischen Elasticsearch und OpenSearch einordnen
- Verstehen, wie Daten in Elasticsearch importiert und indiziert werden
- Erklären, warum Mappings für Analysen und Aggregationen entscheidend sind

Teil 1: Allgemeines zu Suchmaschinen

Warum gibt es Suchmaschinen?

Stell dir vor, du hast ein Buch mit 500 Seiten und suchst alle Stellen, in denen "Umsatz" vorkommt.

Ohne Index: Du blätterst jede Seite durch. Das dauert ewig.

Mit Index (Stichwortverzeichnis): Du schlägst hinten nach und findest sofort alle Seitenzahlen.

> Genau so funktioniert eine Suchmaschine: Sie baut vorab einen Index auf, damit Suchen blitzschnell gehen -- egal wie viele Daten vorhanden sind.

Volltextsuche vs. Datenbankabfrage

Merkmal	Datenbank (SQL)	Suchmaschine
Stärke	Exakte Abfragen	Textsuche, Relevanz
Beispiel	`WHERE stadt = 'Köln'`	"Köln" findet auch "Kölner"
Geschwindigkeit bei Text	Langsam bei großen Mengen	Sehr schnell
Tippfehler	Kein Ergebnis	Trotzdem Treffer möglich
Sortierung	Nach Spalte	Nach Relevanz

> **Merke:** Eine Suchmaschine ersetzt keine Datenbank -- sie ergänzt sie. Viele Unternehmen nutzen beides zusammen.

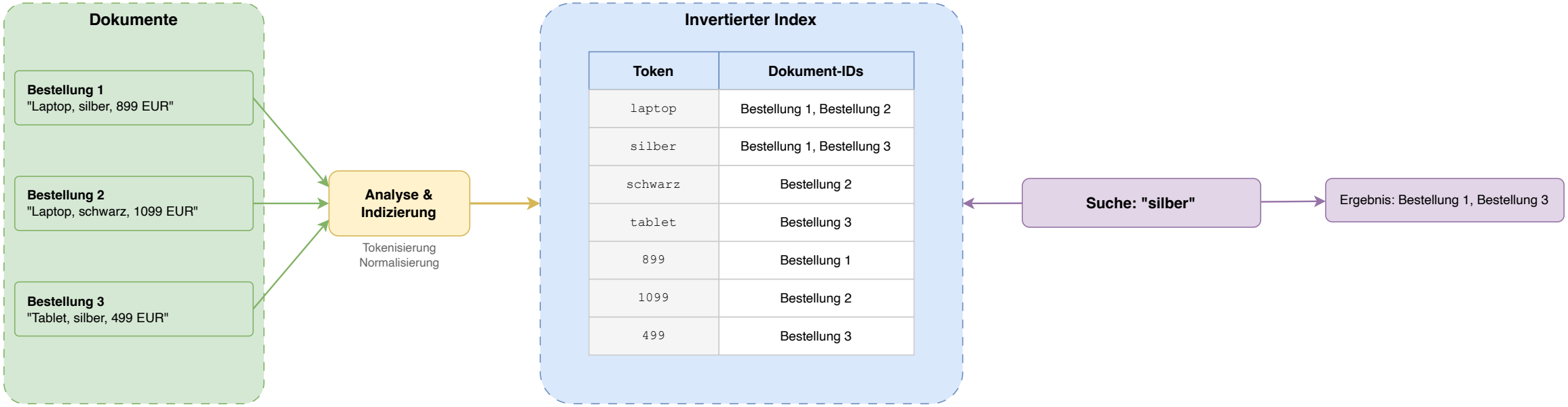
Was ist ein invertierter Index?

Ein invertierter Index dreht die Logik um:
Statt "Welche Wörter enthält Dokument X?"
fragt er "In welchen Dokumenten kommt Wort Y vor?"

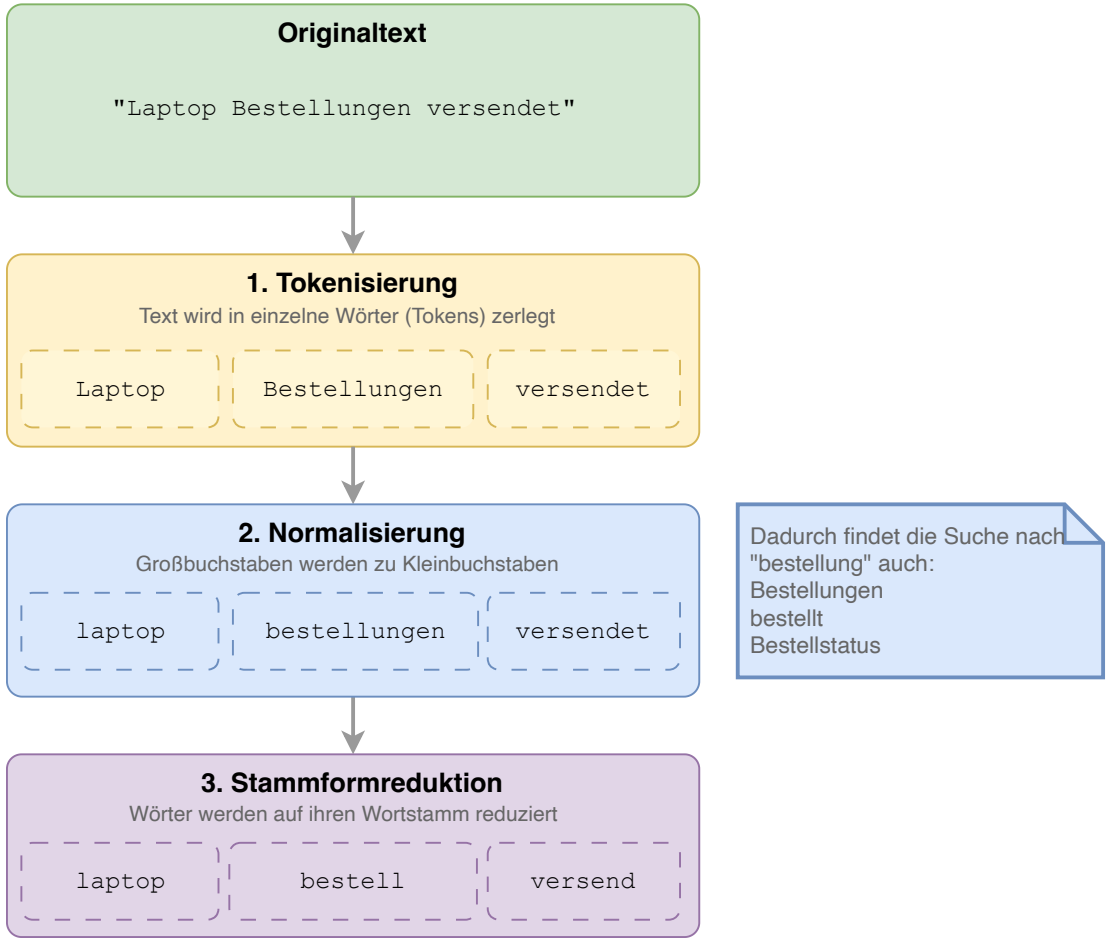
Beispiel - Bestelldaten der Mustertech GmbH:

Dokument	Inhalt
Bestellung 1	"Laptop, silber, 899 EUR"
Bestellung 2	"Laptop, schwarz, 1099 EUR"
Bestellung 3	"Tablet, silber, 499 EUR"

Funktionsweise des invertiertern Index



Prozess der Indexierung



Teil 2: Elasticsearch Grundlagen

Was ist Elasticsearch?

Elasticsearch ist eine **verteilte Such- und Analyseplattform**, die auf dem invertierten Index aufbaut.

Kernmerkmale:

- **Open Source** (mit Einschränkungen, dazu später mehr)
- **Dokumentenorientiert:** Daten werden als JSON-Dokumente gespeichert
- **Schnell:** Auch Milliarden von Datensätzen werden in Sekundenbruchteilen durchsucht
- **Skalierbar:** Wächst mit deinen Daten mit
- **REST-API:** Kommunikation über Standard-HTTP

Was ist ein JSON-Dokument?

JSON ist ein einfaches Textformat zur Darstellung von Daten. Jeder Datensatz in Elasticsearch ist ein JSON-Dokument:

```
{
  "bestellnummer": "MT-2024-001",
  "kunde": "Firma Müller",
  "produkt": "Laptop Pro 15",
  "preis": 1299.00,
  "anzahl": 2,
  "bestelldatum": "2024-03-15",
  "status": "versendet"
}
```

> Keine Zeilen und Spalten wie in Excel -- stattdessen Schlüssel-Wert-Paare.
> Das ist flexibler, weil jedes Dokument unterschiedliche Felder haben kann.

Dokumente, Indizes und Felder

Die wichtigsten Begriffe im Überblick:

Begriff	Erklärung	Analogie
Dokument	Ein einzelner Datensatz	Eine Zeile in Excel
Index	Sammlung von Dokumenten	Ein Excel-Arbeitsblatt
Feld	Eigenschaft eines Dokuments	Eine Spalte in Excel
Mapping	Beschreibung der Felder	Spaltenüberschrift + Typ

Beispiel Mustertech GmbH:

- Index `bestellungen` enthält alle Bestelldokumente
- Index `produkte` enthält alle Produktdokumente
- Index `kunden` enthält alle Kundendokumente

REST-API

Elasticsearch kommuniziert über HTTP -- das ist das gleiche Protokoll, das dein Browser nutzt.

Was heißt das für die Nutzer?

- Du musst keine API-Aufrufe schreiben
- **Kibana** übernimmt die Kommunikation für dich
- Im Hintergrund sendet Kibana Anfragen an Elasticsearch und zeigt dir die Ergebnisse an

Teil 3: Zusammenspiel des Elastic Stack

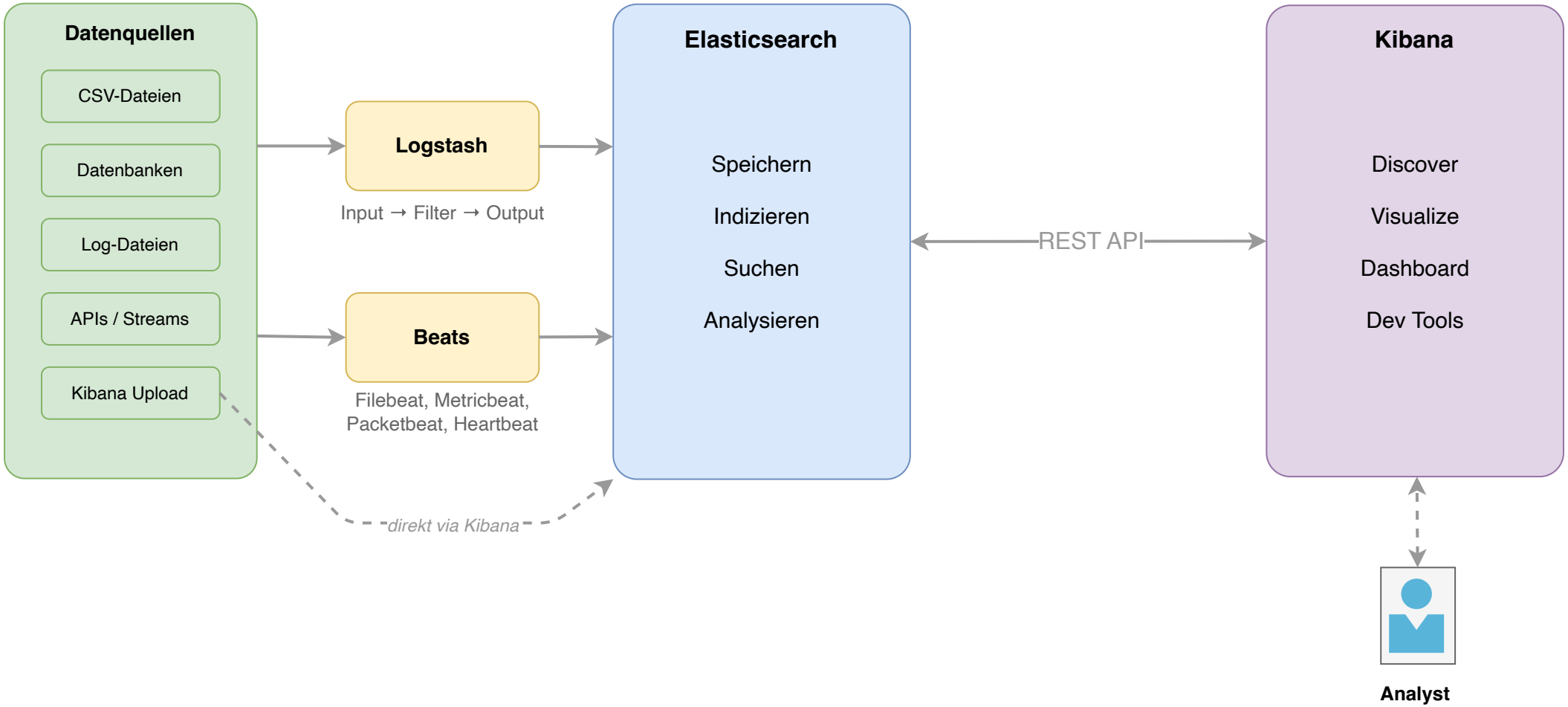
Die vier Komponenten

Der Elastic Stack besteht aus vier Hauptkomponenten:

Komponente	Aufgabe
Elasticsearch	Speichern, Suchen, Analysieren
Kibana	Visualisierung und Bedienoberfläche
Logstash	Daten einlesen und transformieren
Beats	Leichtgewichtige Datensammler

> Früher hieß das Ganze "ELK Stack" (Elasticsearch, Logstash, Kibana). Mit der Ergänzung von Beats wurde es zum "Elastic Stack".

Elastic Stack



Elasticsearch - Der Motor

Elasticsearch ist das Herzstück des Stacks:

- **Speichert** alle Daten als JSON-Dokumente
- **Indiziert** die Daten für schnelle Suche
- **Beantwortet** Suchanfragen und Aggregationen
- **Skaliert** automatisch über mehrere Server

> Elasticsearch arbeitet im Hintergrund. Du interagierst damit über Kibana.

Kibana

Kibana ist die Komponente, mit der man in der Datenanalyse am meisten arbeitet:

- **Discover:** Daten durchsuchen und filtern
- **Visualize:** Diagramme und Grafiken erstellen
- **Dashboard:** Mehrere Visualisierungen kombinieren
- **Dev Tools:** Direkte Anfragen an Elasticsearch
  (für Fortgeschrittene)

> In den nächsten Modulen wirst du Kibana intensiv kennenlernen. Heute schauen wir uns zunächst an, wie die Daten überhaupt nach Elasticsearch gelangen.

Logstash

Logstash ist ein Werkzeug zur Datenverarbeitung:

- **Input:** Daten aus verschiedenen Quellen einlesen (Dateien, Datenbanken, APIs)
- **Filter:** Daten umwandeln, anreichern, bereinigen
- **Output:** Aufbereitete Daten an Elasticsearch senden

Beispiel Mustertech GmbH:
Logstash liest jede Nacht die neuen Bestellungen aus dem ERP-System und schreibt sie in den Elasticsearch-Index `bestellungen`.

> Logstash wird typischerweise von der IT-Abteilung eingerichtet.

Beats

Beats sind leichtgewichtige Programme, die auf Servern oder Arbeitsplätzen laufen und Daten einsammeln:

Beat	Sammelt
Filebeat	Log-Dateien
Metricbeat	System- und Service-Metriken
Packetbeat	Netzwerkdaten
Heartbeat	Verfügbarkeitsdaten

> Für die Mustertech GmbH sammelt Metricbeat z. B. die Antwortzeiten des Online-Shops. Wenn der Shop langsam wird, siehst du das sofort im Kibana-Dashboard.

Teil 4: Unterschiede zwischen Elasticsearch und OpenSearch

Die Geschichte der Trennung

- **Bis 2021:** Elasticsearch war vollständig unter der Apache-2.0-Lizenz (Open Source)
- **Januar 2021:** Elastic (die Firma) änderte die Lizenz auf SSPL und Elastic License -- bestimmte Cloud-Nutzungen waren damit eingeschränkt
- **April 2021:** Amazon (AWS) erstellte einen Fork namens **OpenSearch** unter der Apache-2.0-Lizenz
- **Seitdem:** Zwei getrennte Produkte, die sich unabhängig weiterentwickeln

> **Warum ist das relevant?** Je nach Unternehmen und Cloud-Anbieter triffst du auf das eine oder das andere Produkt.

Wann welches Produkt?

Elasticsearch / Kibana wählen, wenn:

- Du Elastic Cloud nutzen möchtest
- Du Wert auf die neuesten Features legst
  (z. B. ES|QL, ML-Funktionen)
- Dein Unternehmen bereits Elastic-Lizenzen hat

OpenSearch wählen, wenn:

- Du auf AWS arbeitest und den Managed Service nutzen möchtest
- Eine Apache-2.0-Lizenz erforderlich ist
- Du keinen Vendor-Lock-in möchtest

> In dieser Schulung verwenden wir **Elasticsearch und Kibana**.

Teil 5: Datenimport und Indizierung

Wie kommen Daten in Elasticsearch?

Es gibt mehrere Wege, Daten in Elasticsearch zu laden. Hier die wichtigsten:

Weg	Beschreibung	Typischer Nutzer
Kibana Upload	CSV/JSON direkt hochladen	Analyst
Logstash	Automatisierte Pipeline	IT-Abteilung
Beats	Laufende Datensammlung	IT-Abteilung
REST-API	Programmatischer Zugriff	Entwickler
Ingest Pipelines	Verarbeitung beim Import	IT-Abteilung

Kibana File Upload

Mit dem Kibana File Upload kannst du selbstständig Daten importieren:

Unterstützte Formate:

- CSV (Komma- oder Semikolon-getrennt)
- TSV (Tabulator-getrennt)
- JSON (einzelne Dokumente oder Arrays)
- NDJSON (ein JSON-Dokument pro Zeile)

Ablauf Kibana File Upload

1. In Kibana auf "Upload a file" klicken
2. Datei per Drag & Drop hochladen
3. Kibana erkennt automatisch die Struktur
4. Feldtypen prüfen und ggf. anpassen
5. Index-Namen vergeben und importieren

Kibana File Upload - Grenzen

Der Kibana File Upload ist praktisch, hat aber Einschränkungen:

- **Maximale Dateigröße:** Standardmäßig 100 MB
- **Einmaliger Import:** Keine automatische Aktualisierung
- **Einfache Transformationen:** Keine komplexe Datenaufbereitung

Wann nutze ich was?

Szenario	Empfehlung
Einmalige Analyse einer CSV	Kibana Upload
Täglicher Import aus ERP	Logstash
Laufende Server-Überwachung	Beats
Integration in eigene Software	REST-API

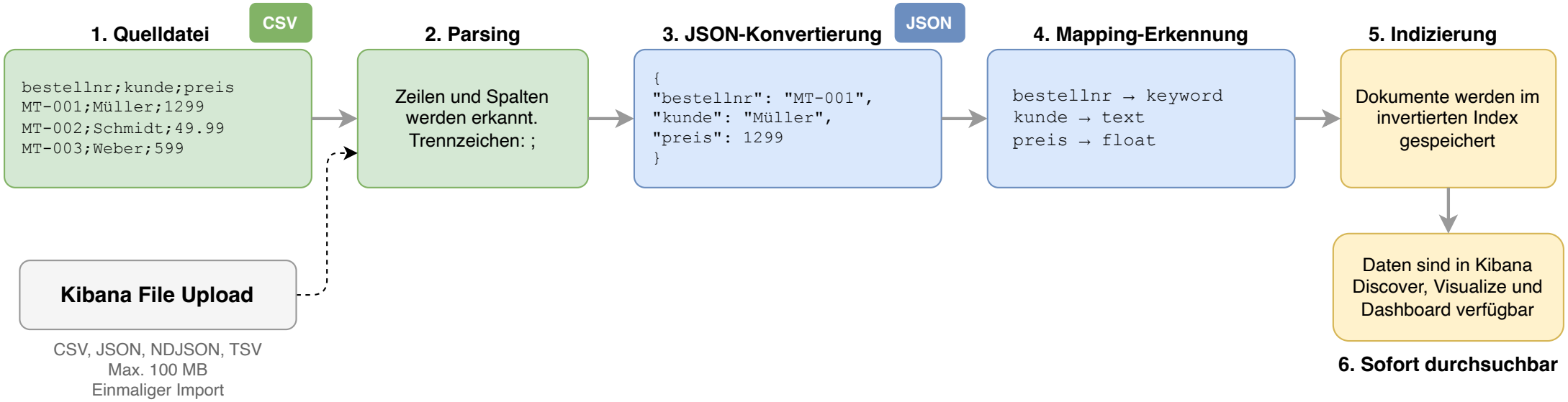
Praxisbeispiel: Bestelldaten importieren

Die Mustertech GmbH exportiert Bestelldaten als CSV-Datei aus dem ERP-System:

```
bestellnummer;kunde;produkt;kategorie;preis;anzahl;datum;status
MT-2024-001;Firma Müller;Laptop Pro 15;Laptops;1299.00;2;2024-03-15;versendet
MT-2024-002;Schmidt AG;USB-C Hub;Zubehör;49.99;10;2024-03-15;versendet
MT-2024-003;Weber & Co;Monitor Ultra 27;Monitore;599.00;3;2024-03-16;in Bearbeitung
MT-2024-004;Firma Müller;Tastatur Ergo;Zubehör;89.99;5;2024-03-16;versendet
```

Diese Datei kann direkt über den Kibana File Upload importiert werden. Kibana erkennt das Semikolon als Trennzeichen und schlägt passende Feldtypen vor.

Prozess des Datenimports



Teil 6: Mappings und Schemafreiheit

Was ist ein Mapping?

Ein Mapping beschreibt die Struktur eines Index -- also welche Felder es gibt und welchen Typ sie haben.

Vergleich mit Excel:

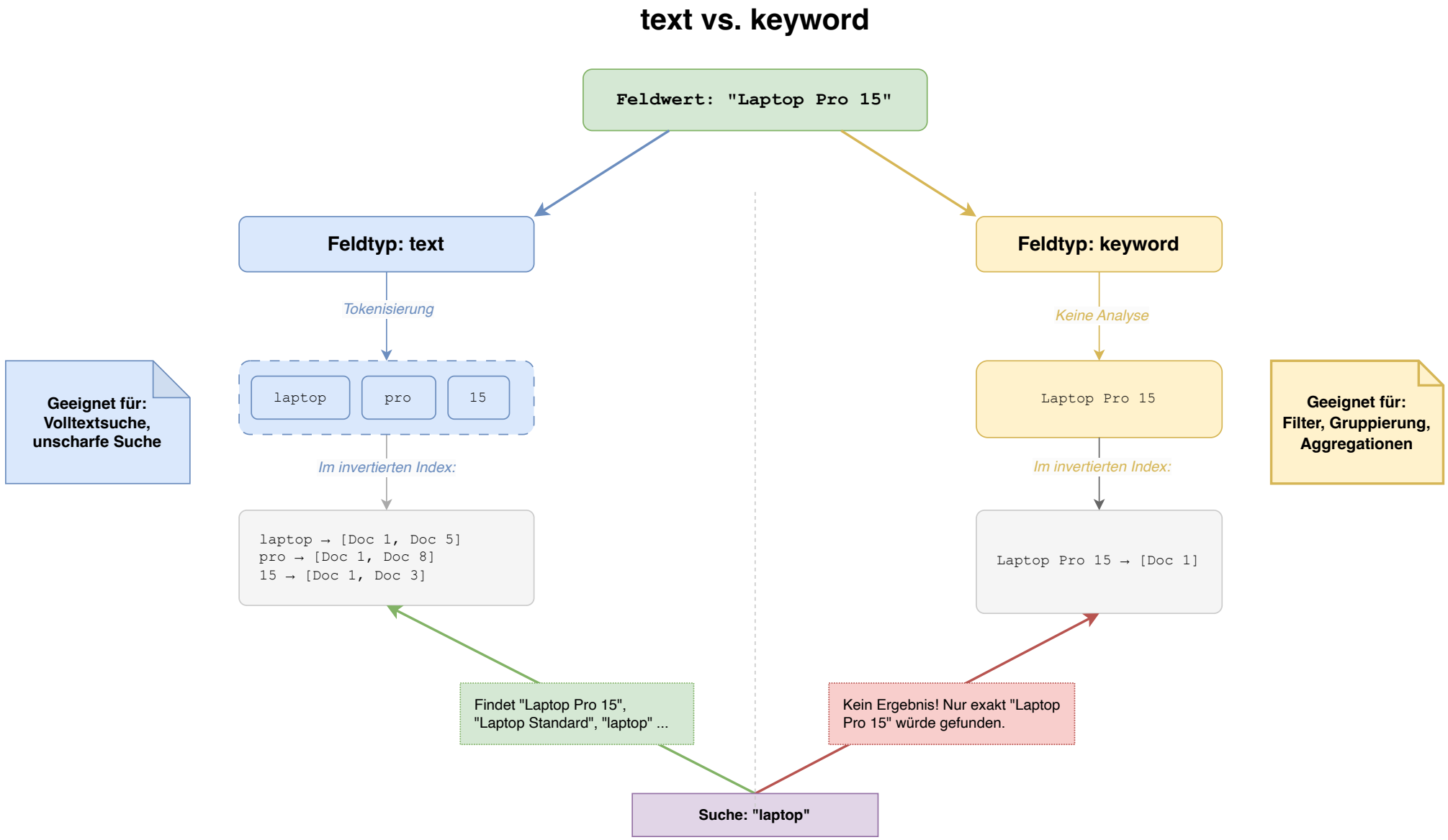
Excel	Elasticsearch
Spalte "Preis" als Währung formatiert	Feld "preis" als Typ `float`
Spalte "Datum" als Datum formatiert	Feld "datum" als Typ `date`
Spalte "Name" als Text	Feld "name" als Typ `text`

> Das Mapping ist entscheidend dafür, was du mit den Daten machen kannst: Ohne richtigen Typ kein korrektes Sortieren, Filtern oder Aggregieren.

Die wichtigsten Feldtypen

Feldtyp	Beschreibung	Beispiel	Nutzung
`text`	Volltextsuche	Produktname	Suchen
`keyword`	Exakter Wert	Status, Kategorie	Filtern, Gruppieren
`long` / `float`	Ganze / Dezimalzahl	Anzahl, Preis	Rechnen, Sortieren
`date`	Datum / Zeitstempel	Bestelldatum	Zeitreihen
`boolean`	Wahr / Falsch	Bezahlt ja/nein	Filtern
`geo_point`	Koordinaten	Lieferadresse	Kartenansicht

> **Tipp:** Achte beim Import besonders auf die Erkennung von `date`-Feldern. Wird ein Datum als `text` erkannt, kannst du keine Zeitreihen-Analysen durchführen.



Schemafreiheit - Fluch und Segen

Elasticsearch ist **schemafrei** (schema-free). Das bedeutet:

Vorteile

- Du kannst Daten importieren, ohne vorher eine Tabellenstruktur festzulegen
- Neue Felder werden automatisch erkannt
- Verschiedene Dokumente im selben Index können unterschiedliche Felder haben

Nachteile

- Automatische Erkennung rät manchmal falsch (z. B. "2024" als Zahl statt als Jahr)
- Einmal gesetzte Feldtypen können nachträglich nicht geändert werden
- Inkonsistente Daten erschweren die Analyse

Warum Mappings für die Analyse

Falsche Feldtypen führen zu Problemen bei der Analyse:

Situation	Problem	Lösung
Preis als `text`	Keine Summe, kein Durchschnitt	Als `float` mappen
Datum als `text`	Keine Zeitreihen möglich	Als `date` mappen
PLZ als `long`	"01234" wird zu "1234"	Als `keyword` mappen
Kategorie als `text`	Gruppierung geht nicht sauber	Als `keyword` mappen

> **Merke:** Prüfe beim Import immer die Feldtypen, bevor du den Import bestätigst. Nachträgliche Änderungen erfordern einen Neu-Import der Daten.

Praxisbeispiel: Mapping der Bestelldaten

So sollte das Mapping für die Bestelldaten der Mustertech GmbH aussehen:

```
{
  "bestellnummer":  { "type": "keyword" },
  "kunde":          { "type": "keyword" },
  "produkt":        { "type": "text" },
  "kategorie":      { "type": "keyword" },
  "preis":          { "type": "float" },
  "anzahl":         { "type": "integer" },
  "datum":          { "type": "date" },
  "status":         { "type": "keyword" }
}
```

- `produkt` als `text`: Volltextsuche nach Produktnamen
- `kategorie` als `keyword`: Gruppierung nach Produktkategorie
- `kunde` als `keyword`: Exaktes Filtern nach Kundennamen

Dynamisches vs. explizites Mapping

Es gibt zwei Wege, das Mapping festzulegen:

Dynamisches Mapping (automatisch)

- Elasticsearch erkennt die Feldtypen selbst
- Praktisch für den schnellen Einstieg
- Risiko: Falsche Typerkennung

Explizites Mapping (manuell)

- Du legst die Feldtypen selbst fest
- Beim Kibana Upload: Feldtypen im letzten Schritt anpassen
- Mehr Kontrolle, weniger Überraschungen

> **Empfehlung:** Lass Elasticsearch die Typen erkennen, aber überprüfe das Ergebnis vor dem finalen Import.

Zusammenfassung

Was wir bisher gelernt haben

- **Suchmaschinen** nutzen einen invertierten Index für schnelle Volltextsuche
- **Elasticsearch** speichert Daten als JSON-Dokumente in Indizes
- Der **Elastic Stack** besteht aus Elasticsearch, Kibana, Logstash und Beats
- **OpenSearch** ist ein Fork von Elasticsearch mit identischen Grundkonzepten
- **Datenimport** für Analysten geht am einfachsten über den Kibana File Upload
- **Mappings** legen fest, welchen Typ ein Feld hat -- das beeinflusst, was du mit den Daten tun kannst

Wichtige Begriffe

Begriff	Bedeutung
Invertierter Index	Verzeichnis: Wort -> Dokumente
Dokument	Ein einzelner Datensatz (JSON)
Index	Sammlung von Dokumenten
Feld	Eigenschaft eines Dokuments
Mapping	Definition der Feldtypen
Tokenisierung	Zerlegung von Text in Einzelwörter
`text`	Feldtyp für Volltextsuche
`keyword`	Feldtyp für exakte Werte

Nächste Schritte

Modul 02: Discover & Abfragen

Im nächsten Modul wirst du:

- Das **Kibana Discover Interface** kennenlernen
- Daten mit der **Kibana Query Language (KQL)** durchsuchen und filtern
- **Zeitfilter** für Zeitreihenanalysen einsetzen
- **Spalten konfigurieren** und Daten exportieren