

PGDay

PostgreSQL als Teil des skalierbaren Open Source
ERP Systems metasfresh

Was ist metasfresh?

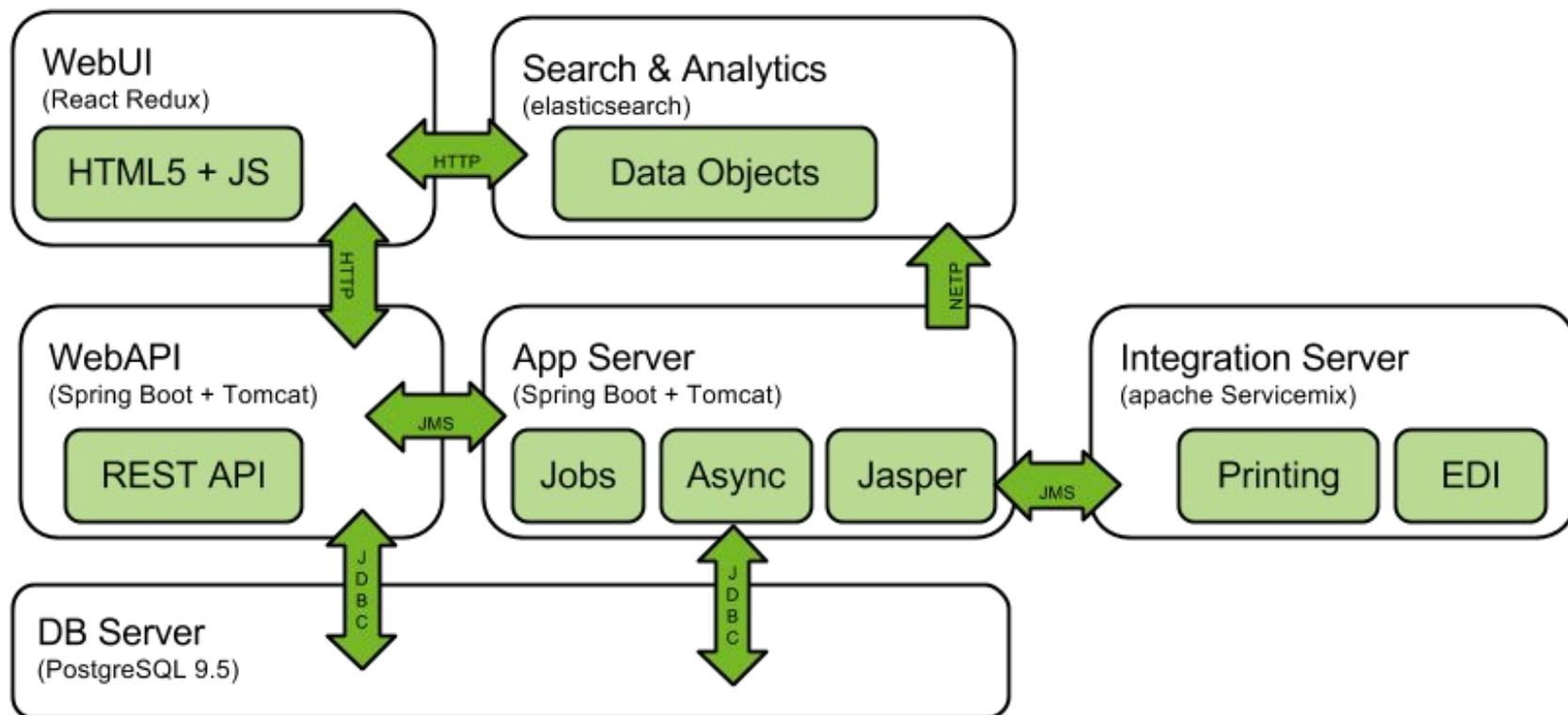
- skalierbare Open Source ERP/CRM Cloud Plattform

Unter **Skalierbarkeit** versteht man die Fähigkeit eines Systems, Netzwerks oder Prozesses zur Größenveränderung. Meist wird dabei die Fähigkeit des Systems zum Wachstum bezeichnet.

Skalierbarkeit

- Datenmenge -

Architektur

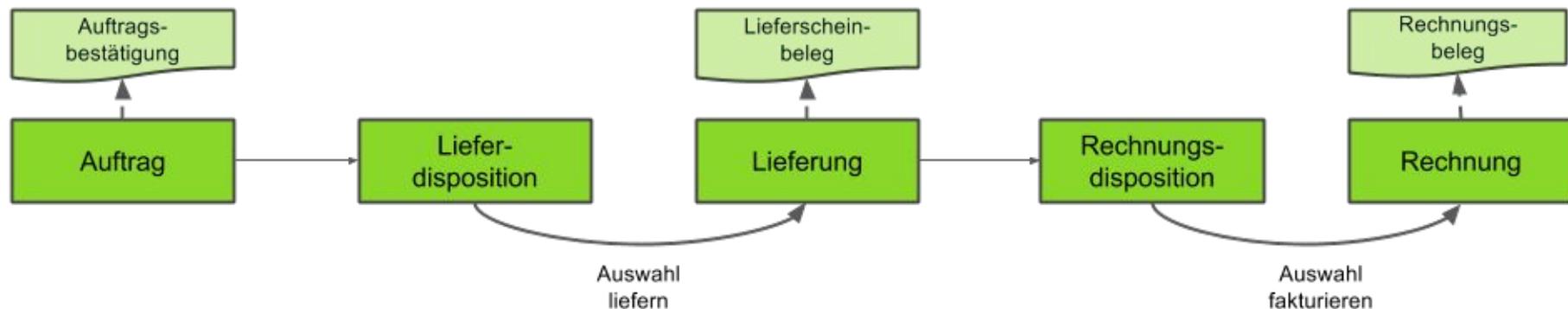


Service	Dev Language	Running on	Technologies	Repo
WebUI	Javascript	Apache / Nginx	React Redux, HTML5, PostCSS	Github
WebAPI	Java 8	Spring Boot	REST, JSON, Swagger, Spring, hazelcast, websocket	Github
App	Java 8, SQL	Spring Boot	Jasper Reports, Application Dictionary	Github
DB	SQL, PostgreSQL	Postgres (9.5+)	Application Dictionary	Github
Reporting	JRXML	Jasperserver	Jasper Reports 6.5.1 Client	Github
elastic Search			Standard Elastic Search	Github
RabbitMQ			Standard RabbitMQ	Github
Java Client	Java 8	Java JRE 8+	Swing	Github

http://docs.metasfresh.org/howto_collection/EN/metasfresh_architecture.html

Skalierbarkeit - Prozesse -

Demo



Skalierbarkeit - Neue Features -

Wöchentliche Releases

Release 5.107

Mark Krake / 14. Juni 2019 /
[Blog, Release notes](#)



Release 5.107 was last modified: Juni
17th, 2019 by Mark Krake

Release 5.106

Mark Krake / 1. Juni 2019 /
[Blog, Release notes](#)



Release 5.106 was last modified: Juni
1st, 2019 by Mark Krake

Release 5.105

Mark Krake / 24. Mai 2019 /
[Blog, Release notes](#)



Release 5.105 was last modified: Mai
24th, 2019 by Mark Krake

Release 5.104

Mark Krake / 17. Mai 2019 /
[Blog, Release notes](#)



Release 5.104 was last modified: Mai
17th, 2019 by Mark Krake

Release 5.102

Mark Krake / 3. Mai 2019 /
[Blog, Release notes](#)



Release 5.102 was last modified: Mai
3rd, 2019 by Mark Krake

Release 5.103

Mark Krake / 11. Mai 2019 /
[Blog, Release notes](#)



Release 5.103 was last modified: Mai
11th, 2019 by Mark Krake

Release 5.99

Mark Krake / 12. April 2019 /
[Blog, Release notes](#)



Release 5.100

Mark Krake / 19. April 2019 /
[Blog, Release notes](#)



Release 5.101

Mark Krake / 29. April 2019 /
[Blog, Release notes](#)



Skalierbarkeit - Customization -

Application Dictionary

UI Rendering

The screenshot shows a web application interface with several annotated components:

- Tab Level 0**: Points to the top navigation bar.
- Menu Node Name**: Points to the metasfresh logo in the top right.
- Field**: Points to the 'Memo' text input area.
- AD Column**: Points to the 'Suchschlüssel' field.
- UI Element Group**: Points to the 'Firma' field.
- UI Column**: Points to the 'Firmenname' field.
- UI Element**: Points to the 'Firmenname' field.
- AD Column: Type Checkbox**: Points to the 'Aktiv' toggle switch.
- AD Column: Type Datum**: Points to the 'Wiedervorlage Dat...' field.
- AD Column: Referenz Multiline Text**: Points to the 'URL' field.
- Tab Level 1**: Points to the 'Vorgänge' tab.
- AD Table**: Points to the table header.
- UI Grid**: Points to the table header.

The main form contains the following fields:

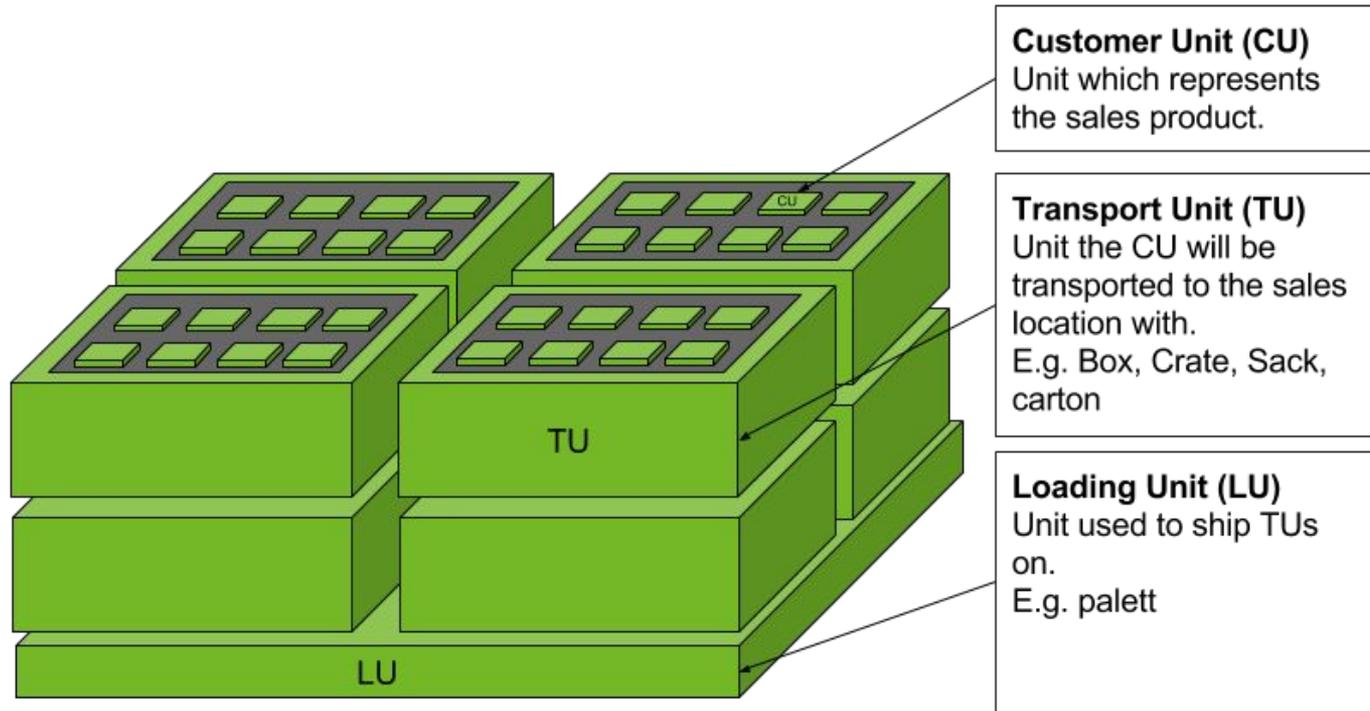
Suchschlüssel	1000000	Aktiv	<input type="checkbox"/>
Firma	<input checked="" type="checkbox"/>	Aussendienst	leer
Firmenname	Neues System GmbH	Wiedervorlage Dat...	leer
Name Zusatz	leer	Innendienst	leer
Memo	<input type="text" value="leer"/>	Wiedervorlage Dat...	leer
		URL	leer

The table below the form is currently empty:

Nr	Vorgangsdatum	Erstellt durch	Vorgangsart	Notiz	Status
Es sind noch keine Detailzeilen vorhanden. Du kannst sie im jeweiligen Fenster erfassen.					

Skalierbarkeit Generisches Modell

metasfresh Handling Unit System



Typische Use Cases

- stark skalierende Unternehmen
 - hohe Belegzahlen (Aufträge, Abos, etc.)
 - hohe Dynamik im Wachstum
- Anforderung nach Rückverfolgbarkeit

Branchen

- Lebensmittel
- Parma
- Maschinenbau
- Dienstleistung
- Verbände
- Verlage

Einsatz von Postgres

- Metriken => siehe Datagrip
- Zugriff von metasfresh via eigenen Datenzugriffslayer
 - Caching
 - Cache Invalidation => Demo
- Installationsformen
 - Dediziert mit Standby Server und streaming Replication
 - Docker Container
- JSON Fields
- DB Größe 30 MB bis 250GB
- andere engines
 - elastic Search, RabbitMQ

Erfahrungen mit Postgres

- in 15 Jahren keinen Datenverlust
- Sehr stabil
- Tuning der indices bei großen Datenmengen aufwendig
- PGAdmin durch Datagrip ersetzt
- Updates bis auf Details sehr gut verlaufen

Ausblick auf die Nutzung von Postgres

- andere, spezialisierte Datenengines kommen
- Backbone der transaktionalen Datenverwaltung wird Postgres bleiben

Links

- Code: <https://github.com/metasfresh>
- Demo: <https://metasfresh.com/nextgen/#Start>
- Download: <http://metasfresh.com/download>
- Forum: <https://forum.metasfresh.com>
- Docs: <https://docs.metasfresh.org>
- Architektur: http://docs.metasfresh.org/.../metasfresh_architecture.html
- Twitter: @metasfresh