Informix Tenants

Ein Erfahrungsbericht

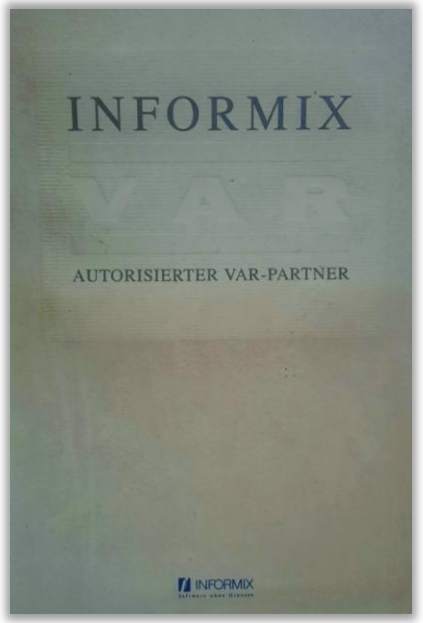


Dr.-Ing. Peter Ijewski

Vorstellung

- Dr.-Ing. Peter Ijewski
- Maschinenwesen an der RWTH Aachen, Promotion am Institut für Regelungstechnik
- Systemspezialist für Logistik in einem Softwarehaus
- Leiter Organisation und Datenverarbeitung in einem weltweit operierenden Ingenieurbüro
- Seit 1992 Eigenes Unternehmen, u. a. Entwicklung und Vertrieb eines ERP Systems
- IT-Consulting & Schulungen
- Bis 2017 div. Lehraufträge an der Dualen Hochschule BW

- Erfahrung mit diversen DBMS
- Informix seit 1989, seit einigen Jahren Informix Design Partner
- Programmiererfahrung seit 1977



Einleitung

Agenda

- Vorstellung
- Tenant-Datenbank – Was ist das?
- Tenant-Datenbank – Wie geht das?
- Tenant-Datenbanken in der Praxis
  - Motivation und Aufgabenstellung
  - Vorgehensweise und Best Practices
  - Aktuelle Situation
- Zusammenfassung
- Fragen

Tenant Datenbank

Tenant - Was ist denn das?



Tenant -> Mieter, Mandant

Tenant Datenbank

Tenant - Was ist denn das? (Quelle: Wikipedia)

Software multitenancy is a software architecture in which a single instance of software runs on a server and serves multiple tenants.

Systems designed in such manner are "shared" (rather than "dedicated" or "isolated"). A tenant is a group of users who share a common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance - including its data, configuration, user management, tenant individual functionality and non-functional properties.

Multitenancy contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants. Some commentators regard multitenancy as an important feature of cloud computing.

Tenant Datenbank

Tenant Datenbank - Was ist denn das?

Ein Datenbank-Tenant ist ein logisch getrenntes (Kunden-)System innerhalb einer gemeinsamen Datenbank oder Anwendung.

Beispiel: Eine Anwendung wird von mehreren Firmen genutzt. Jede Firma ist ein Tenant und sieht nur ihre eigenen Daten.

Typische technische Umsetzungen sind:

1. **Gemeinsame Datenbank, gemeinsame Tabellen** Jede Tabelle enthält z. B. eine `tenant_id`, über die Daten getrennt werden.
2. **Gemeinsame Datenbank, getrennte Schemata** Jeder Tenant erhält ein eigenes Datenbankschema.
3. **Eigene Datenbank pro Tenant** Die Daten sind physisch vollständig getrennt.

Der Begriff wird vor allem bei Multi-Tenant-Systemen verwendet. Wichtig ist, dass die Anwendung die Tenant-Grenzen konsequent erzwingt, damit ein Kunde nicht auf Daten eines anderen Kunden zugreifen kann.

Tenant Datenbank

Informix Tenant - Was ist das?

In IBM Informix wird das Konzept des Tenants über eine spezifische Funktion namens Multi-Tenancy (oft im Zusammenhang mit Cloud-Lösungen) implementiert.

Im Gegensatz zu einer rein (Anwendungs-)softwareseitigen Lösung (wie einer `TenantID`-Spalte in einer Tabelle) bietet Informix eine tiefere Integration auf Datenbankebene.

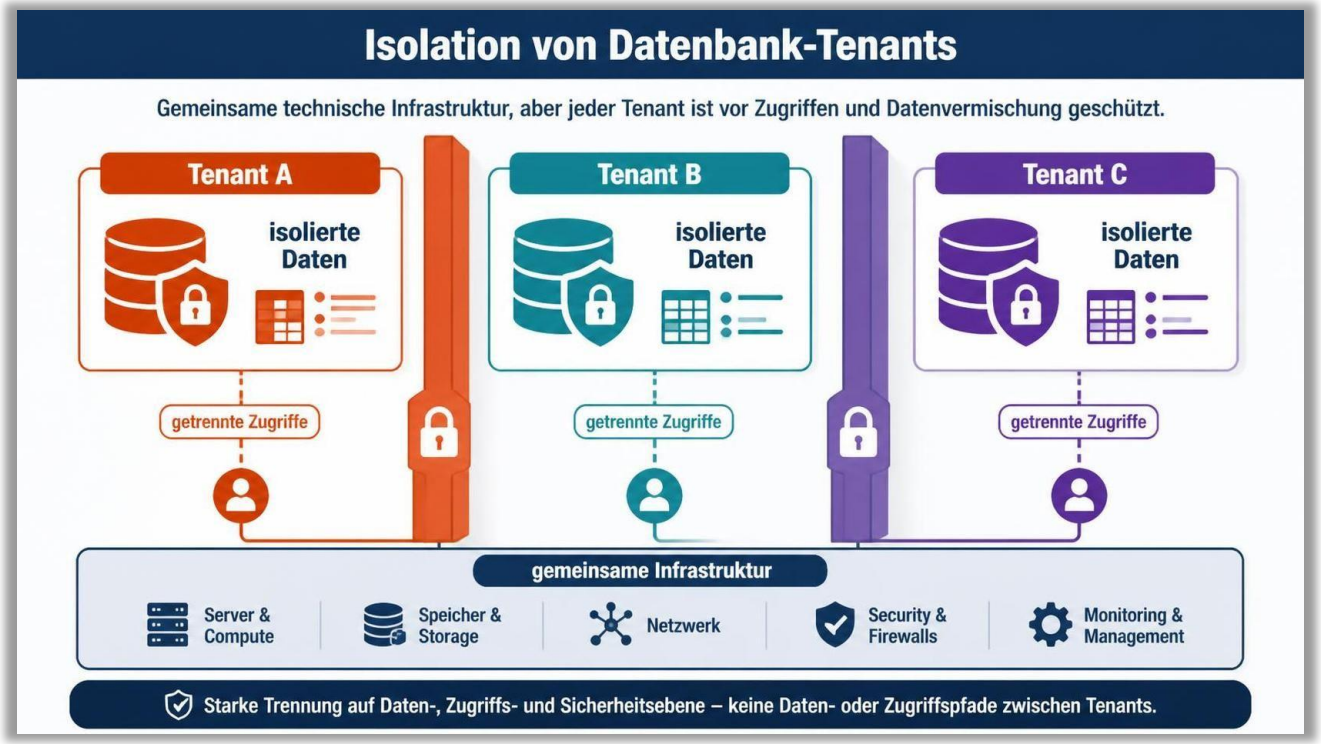
Ein Informix-Tenant ist im Wesentlichen eine logische Partitionierung der Datenbank, die es ermöglicht, mehrere Mandanten in einer einzigen Datenbank-Instanz zu verwalten, während sie sich gegenseitig „unsichtbar" bleiben.

Tenant Datenbank

Spezifische Merkmale eines Informix-Tenants

1. Logische Isolation (Tenant-ID)
   Informix nutzt intern eine Tenant-ID, um Daten zu markieren. Das Besondere ist, dass diese Trennung transparent für die Anwendung erfolgt. Wenn eine Verbindung zu der (Tenant-) Datenbank hergestellt wird, wird der Kontext transparent auf einen spezifischen Tenant gesetzt.

   Alle darauf folgenden Abfragen greifen dann automatisch nur auf die Daten dieses Tenants zu. Konstrukte wie `WHERE tenant_id = 'XYZ'` sind überflüssig.



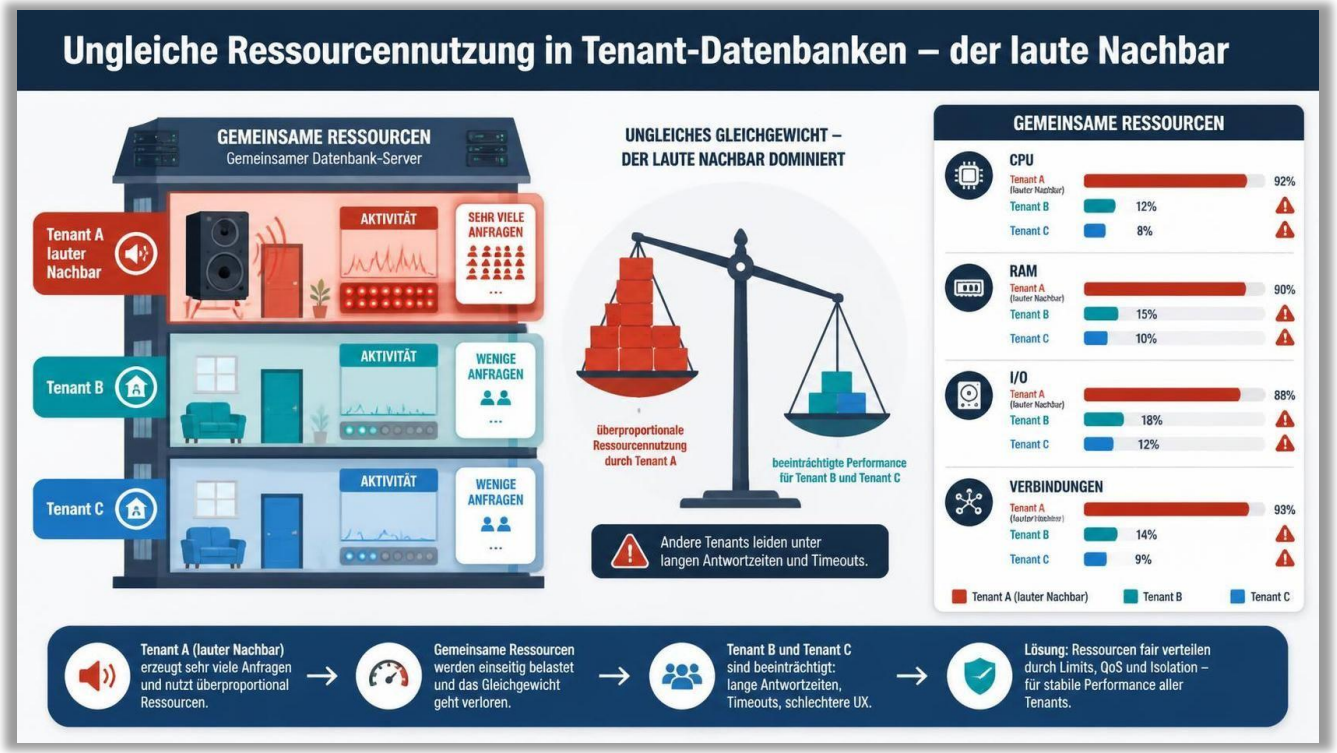
KI-Generiert

Tenant Datenbank

Spezifische Merkmale eines Informix-Tenants

2. Ressourcenmanagement
   Ein großer Vorteil bei Informix ist, dass Tenants nicht nur logisch getrennt sind, sondern auch die Ressourcenzuordnung und -verwendung gezielt gesteuert werden können.

   Man kann z. B. festlegen, wie viel Speicher oder Rechenleistung einem bestimmten Tenant zur Verfügung steht, um zu verhindern, dass ein „lauter Nachbar" (also ein Tenant mit extrem hoher Last) die Performance für alle anderen Mandanten beeinträchtigt.



KI-Generiert

Tenant Datenbank

Spezifische Merkmale eines Informix-Tenants

3. Verwaltung und Sicherheit
   Ein Informix-Tenant ermöglicht eine granulare Zugriffskontrolle. Administratoren können Benutzer so zuweisen, dass sie nur innerhalb ihres zugewiesenen Tenants operieren.

   Dies vereinfacht die Compliance und die Datensicherheit erheblich, da die Trennung tief in der Engine der Datenbank verankert ist und nicht nur auf einer oberflächlichen Anwendungsschicht existiert.



KI-Generiert

Tenant Datenbank

Spezifische Merkmale eines Informix-Tenants

Fazit: Ein Informix-Tenant ist ein „logisch abgegrenzter Mandantenbereich".

Er erlaubt es, eine einzige Informix-Installation so zu betreiben, dass sie sich

- für den Endnutzer und die Anwendung wie eine private, dedizierte Datenbank anfühlt,
- und der Betreiber die Effizienzvorteile einer gemeinsam genutzten Infrastruktur nutzt.

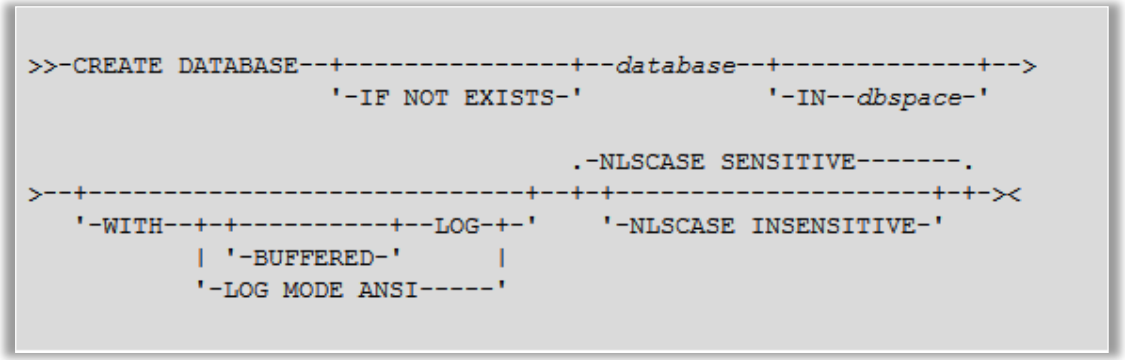


KI-Generiert

Tenant Datenbank

Tenant-Datenbank – Wie geht das?

Klassische Syntax zum Anlegen einer Datenbank:



„Neue" Syntax zum Anlegen einer Tenant-Datenbank

```
EXECUTE FUNCTION task('tenant create','database_name',
                      '{dbspace:"datendbs, indexdbs",
                        sbspace:"blobdbs",
                        logmode:"unbuffered"
                        case:"sensitive",
                        locale:"en_US.819",
                        session_limit_locks:"10000",
                        session_limit_logspace:"5 GB",
                        …
}'
                     );
```

Tenant Datenbank

Tenant-Datenbank – Wie geht das?

Syntax zum Ändern eines Tenant

```
EXECUTE FUNCTION task('tenant update','database_name',
                      '{session_limit_txn_time:"300",
                        …
                       }'
                     );
```

Syntax zum Entfernen eines Tenant

```
EXECUTE FUNCTION task('tenant drop','database_name');
```

Tenant Datenbank

Tenant-Datenbank – Warum denn das?

Motivation für Multitenancy mit Informix
Bei einer Multitenancy-Lösung auf Basis von IBM Informix nutzen mehrere Mandanten dieselbe Informix-Plattform, bleiben aber logisch oder physisch voneinander getrennt. Ein Mandant kann beispielsweise ein Kunde, eine Behörde, eine Krankenkasse oder eine Einrichtung sein.

Zentrale Ziele sind:

- Datentrennung nach DSGVO:
  Daten müssen so getrennt werden, dass personenbezogene Informationen eines Mandanten nicht versehentlich von einem anderen Mandanten eingesehen oder verarbeitet werden können.

- Berücksichtigung von SGB-Anforderungen:
  Im sozialrechtlichen Umfeld sind insbesondere Vertraulichkeit, Zugriffsbeschränkung, Nachvollziehbarkeit und der Schutz besonders sensibler Sozialdaten relevant.

Tenant Datenbank

Tenant-Datenbank – Warum denn das?

- Wirksame Tenant-Isolation:
  Die Trennung muss nicht nur auf Anwendungsebene, sondern auch in Informix, bei Schnittstellen, Jobs, Backups, Exporten, Caches und Administrationsprozessen durchgesetzt werden.
- Bessere Ressourcennutzung:
  CPU, Arbeitsspeicher, Storage, Informix-Instanzen und Betriebspersonal können gemeinsam genutzt werden. Dadurch werden Ressourcen gleichmäßiger ausgelastet.
- Zentrale Wartung:
  Informix-Versionen, Sicherheitsupdates, Backup-Verfahren, Monitoring und Betriebsprozesse lassen sich zentral verwalten.
- Skalierbarkeit:
  Neue Tenants können standardisiert angelegt werden, ohne jedes Mal eine vollständig neue Systemlandschaft aufzubauen.
- Nachvollziehbarkeit:
  Zugriffe und Änderungen sollten eindeutig einem Tenant, Benutzer und Vorgang zugeordnet werden können.

Tenant Datenbank

Tenant-Datenbank – Warum nicht gleich in der Cloud?

Vorteile des On-Premise-Einsatz:

- Hohe Kontrolle über Server, Netzwerk, Storage und Sicherheitsmaßnahmen.
- Daten können innerhalb der eigenen Infrastruktur oder eines definierten Rechtsraums verbleiben.
- Bestehende Informix-Systeme und individuelle Betriebsprozesse können weiter genutzt werden.
- Eigene Vorgaben für Backup, Verschlüsselung, Zugriff und Notfallwiederherstellung lassen sich direkt umsetzen.
- Eine physische oder virtuelle Trennung von Tenants ist leichter organisatorisch kontrollierbar.

Nachteile (oder ebenfalls Vorteile?):
Die Organisation trägt Verantwortung für Beschaffung, Betrieb, Wartung, Patching, Monitoring und Ausfallsicherheit.

Tenant Datenbank

Tenant-Datenbank – Warum nicht gleich in der Cloud?

Vorteile Cloud-Einsatz:
- Rechenleistung und Storage können leichter skaliert werden.
- Betriebskosten können stärker an der tatsächlichen Nutzung ausgerichtet werden.

Nachteile und zusätzliche Anforderungen:
- Für SGB-relevante Daten sind besonders strenge Anforderungen an Zugriffsschutz, Protokollierung, Verschlüsselung, Auftragsverarbeitung und Nachweisbarkeit zu beachten.

- Der US CLOUD Act erlaubt amerikanischen Behörden Zugriff auf Daten weltweit — auch wenn sie auf EU-Servern liegen. –> Showstopper!

Tenant Datenbank

Ursprüngliche Aufgabenstellung

Aufgabenstellung:
- 40 Entwickler
- 50+ Entwicklerdatenbanken
- limitiere Ressourcen, (z. B. in der Developer-Lizenz)

Entwickler tendieren bei begrenzten Ressourcen meist dazu,
- Speicher- und Rechenaufwand zu reduzieren,
- effiziente Algorithmen und Datenstrukturen einzusetzen,
- usw.

In der Realität wird jedoch häufig über die Einschränkungen geschimpft – etwa über zu wenig Speicher, Rechenleistung, Zeit oder Budget –, anstatt die vorhandenen Mittel konsequent effizient zu nutzen. Restriktive Rahmenbedingungen führen **leider** nicht automatisch zu einem bewussteren oder verantwortungsvolleren Umgang mit begrenzten Ressourcen.

Tenant Datenbank

Ursprüngliche Aufgabenstellung

Ergo: Positive Effekte hervorheben!

Enge Vorgaben nach SGB zur Datentrennung

Bisher : Isolation durch unterschiedliche Server/Instanzen und ggf. DBMS-Vendor
Zukünftig: Isolation durch unterschiedliche Tenants

Proaktiver Schutz der Instanzen gegen Fehlbedienungen, etc.

Bisher : ggf. Probleme durch lange Transaktionen, falsche Extentsizes, zu viele Sperren. falsche Isolationslevel, u. a.
Zukünftig: Bessere Einflussnahme durch Kontingentierung

Definierter Restore einer einzelnen Datenbank (Fehlinterpretation: Sicherung vs. Archivierung)

Bisher : aufwändig, ggf. "archecker"
Zukünftig: Komfortable, simple Lösung durch tenant-Restore

Tenant Datenbank

Vorgehensweise Migration




```
$ dbexport database_name -nw -ss -o outdir
```

Ggf. Detail-Anpassungen in

`outdir/database_name.sql` (Tablespaces, etc.)

```
$ dbimport -exists tenant_database -d datendbs -i indir
```

Tenant Datenbank

Vorgehensweise Neuerstellung

Neuerstellung durch parametrisierbares Skript (Prinzipdarstellung, Auszug)

```
  cd /dbspace
  touch $1_datendbs.000
  touch $1_indexdbs.000
# Rechtevergabe
  chmod 660 $1_datendbs.000
  chmod 660 $1_indexdbs.000
  chown informix:informix $1_datendbs.000
  chown informix:informix $1_indexdbs.000
  echo "$(date) Anlegen von  $1_datendbs mit einer Größe von $2 MB"
  sizeDBS=`expr $2 \\* 1024`
  onspaces -c -d $1_datendbs -p /dbspace/$1_datendbs.000 -o 0 -s $sizeDBS
  echo "$(date) Anlegen von  $1_indexdbs mit einer Größe von $3 MB"
  sizeIDX=`expr $3 \\* 1024`
  onspaces -c -d $1_indexdbs -p /dbspace/$1_indexdbs.000 -o 0 -s $sizeIDX
  …
#
$INFORMIXDIR/bin/dbaccess sysadmin  <<EOSQL
  EXECUTE FUNCTION task('tenant create', '$1',
                        '{dbspace:" $1_datendbs, $1_indexdbs",
                          logmode:"unbuffered"}');
EOSQL
```

Tenant Datenbank

Monitoring

```
$ INFORMIXDIR/bin/dbaccess sysadmin  <<EOSQL
   select tenant_dbsname,
          tenant_create_time,
          tenant_last_updated,
          tenant_resources::json as info
     from tenant
 order by tenant_dbsname;
EOSQL

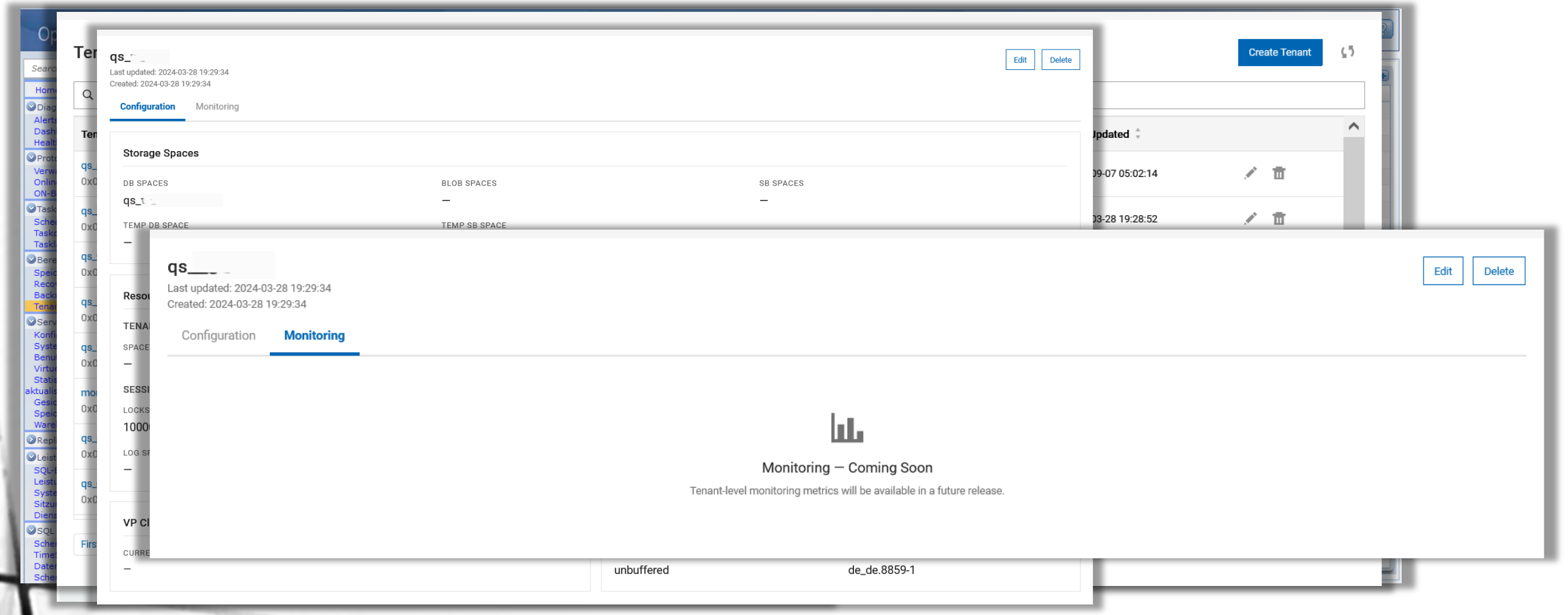
tenant_dbsname       test_
tenant_create_time   2026-
tenant_last_updat+   2026-
info                 {"dbs
                     _limi
```

```
$ onstat -g cac all
.
.

        ID 9       Ordval 70        TxID 0        Flags Active  Hits 7          ucount 0
Last access 06/06/2026.13:11
        database                : test_ije 4600002
        dbspace                 : test_ije_datendbs,test_ije_indexdbs,
        session_limit_locks     : 1000
        memory_limit            : 0.00 MB
        used_memory             : 0.00 MB
        connection_limit        : 0
        used_connections        : 0
.
.
```

Tenant Datenbank

Monitoring



Tenant Datenbank

Tenant restore

```
$ onbar -r -T test_ije -t "2026-06-06 14:30:00" -O


13:28:06  Level 0 Archive started on test_ije_dbs
13:28:07  Archive on test_ije_dbs Completed.
13:28:08  Logical Log 4698 Complete, timestamp: 0x1d0e90d2.
13:28:08  Tenant PIT restore process finished (test_ije: rc=0, tenant_state=ok).
13:28:08  Update of tenant_state for tenant database (test_ije) succeeded.
```

Siehe auch Informix-Newsletter Januar 2016
Gerd Kaluzinski: „Restore einer einzelnen Datenbank aus einem binären Backup“

Tenant Datenbank

Aktuelle Situation

10 Jahre später und einer fast vollständigen Umstellung von Einzeldatenbanken zu Tenants:

Deutlich geringere Betriebskosten:
Mehrere Tenants nutzen gemeinsame Informix-Instanzen, Server, Storage- und Netzwerkressourcen. Dadurch sinken Investitions-, Lizenz- und Wartungskosten gegenüber vielen vollständig getrennten Systemen.

Bessere Ressourcenauslastung:
Freie Kapazitäten eines Tenants können von anderen Tenants genutzt werden. Leistung und Speicher lassen sich dadurch effizienter auslasten.

Vereinfachte Wartung:
Updates, Sicherheitskorrekturen, Monitoring und zentrale Konfigurationsänderungen müssen nicht für jede einzelne (Kunden-)Umgebung separat umgesetzt werden.

Tenant Datenbank

Aktuelle Situation

Nachweisbare Datentrennung:
Über die Jahre können etablierte Prüfverfahren, Tenant-Kennungen, Berechtigungskonzepte und technische Kontrollen die Einhaltung von DSGVO- und gegebenenfalls SGB-Anforderungen unterstützen.

Effizientere Backups und Wiederherstellung:
Gemeinsame, standardisierte Backup- und Restore-Verfahren reduzieren den administrativen Aufwand. Gleichzeitig können Verfahren zur Wiederherstellung einzelner Tenants weiterentwickelt werden.

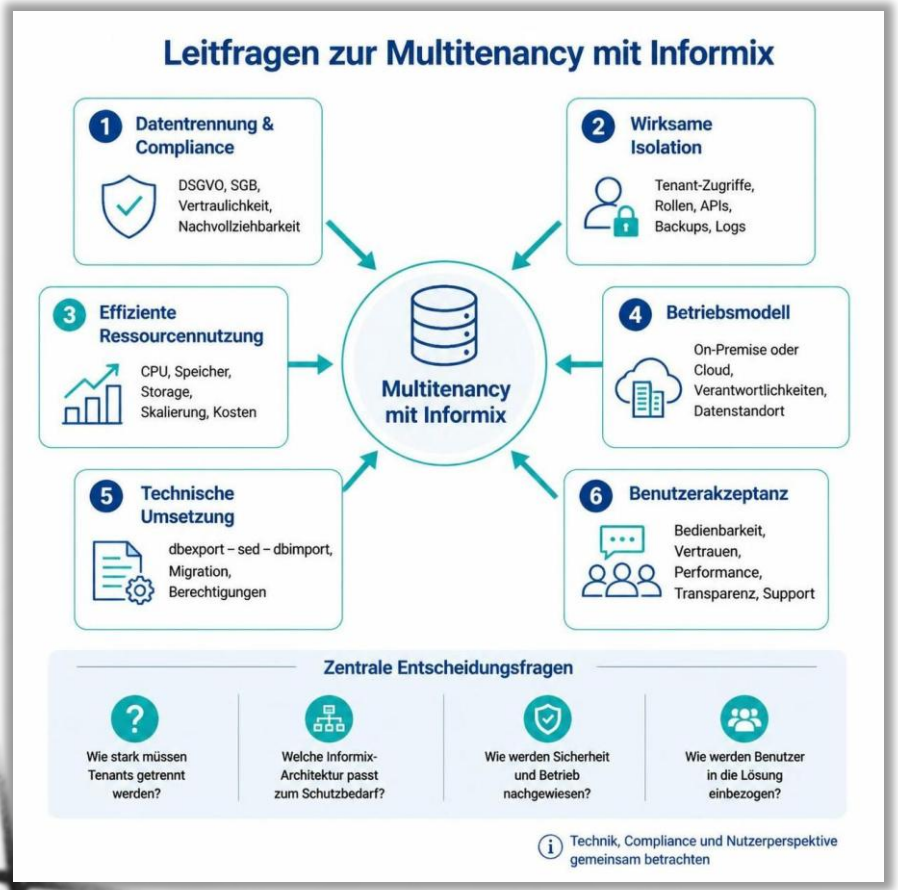
Höhere Benutzerakzeptanz:
…



KI-Generiert

Tenant Datenbank

Zusammenfassung





KI-Generiert

Informix Tenants

Vielen Dank

