



Indizies In Depth



Thorsten Kansy (tkansy@dotnetconsulting.eu)

Meine Person- Thorsten Kansy

Freier Consultant, Software Architekt,
Entwickler, Trainer & Fachautor



Azure Cosmos DB

Mein Service- Ihr Benefit

- Individuelle Inhouse Trainings
- (Online on-demand) Projektbegleitung
- Beratung
 - Problemanalyse und Lösungen
 - Technologieentscheidungen



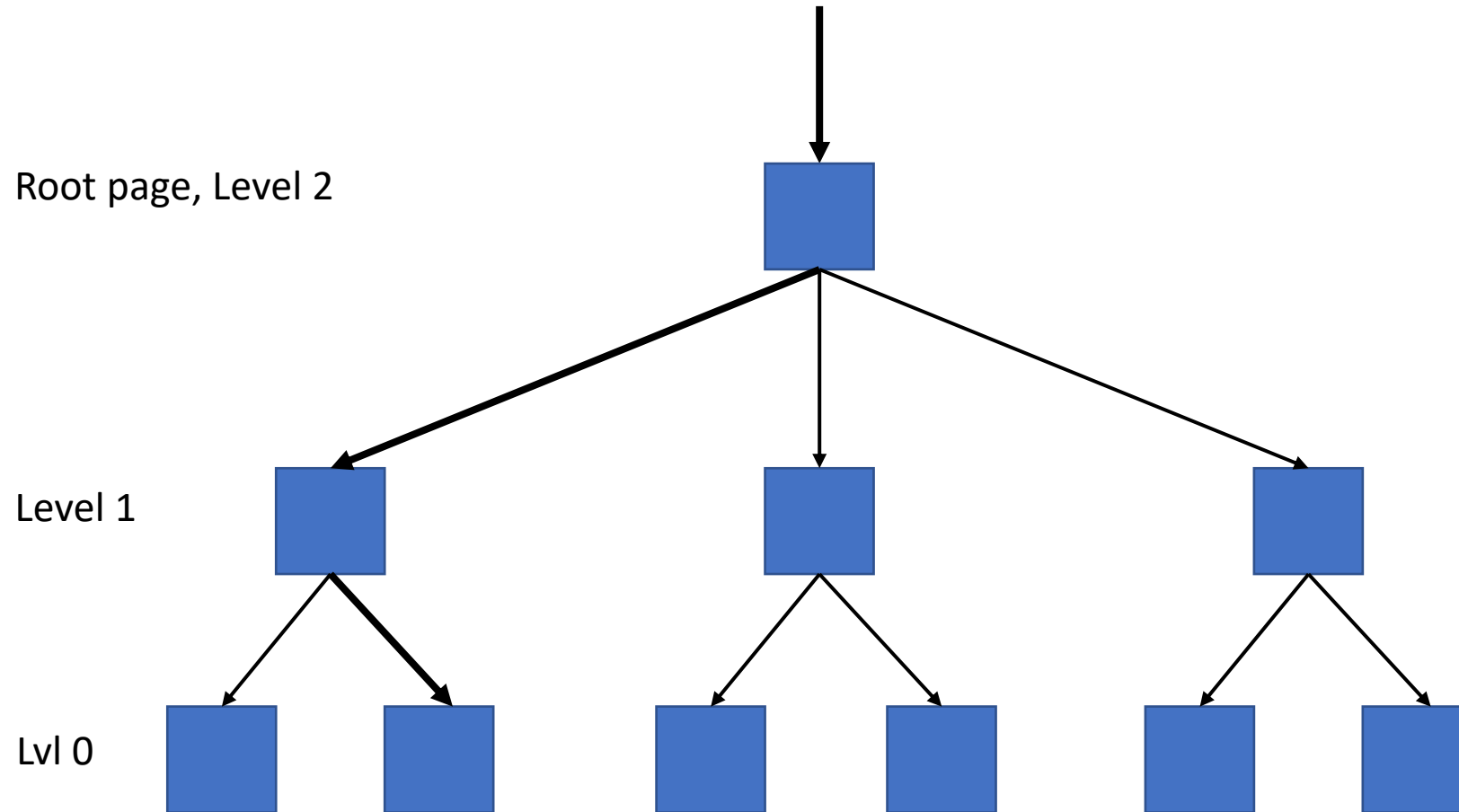
Agenda

- Interne Funktionsweise
- Indizes & die SQL Server Storage Engine
- Page splits
- Fragmentation
- FILLFACTOR/ PAD_INDEX
- Abhilfe?

Indizes

- Clustered Index
 - Daten auf Leaf-Level
 - Navigation auf non Leaf-Level
 - B(alanced)-Tree
- Nonclustered index
 - Daten auf Leaf-Level
 - Navigation auf non Leaf-Level
 - B(alanced)-Tree

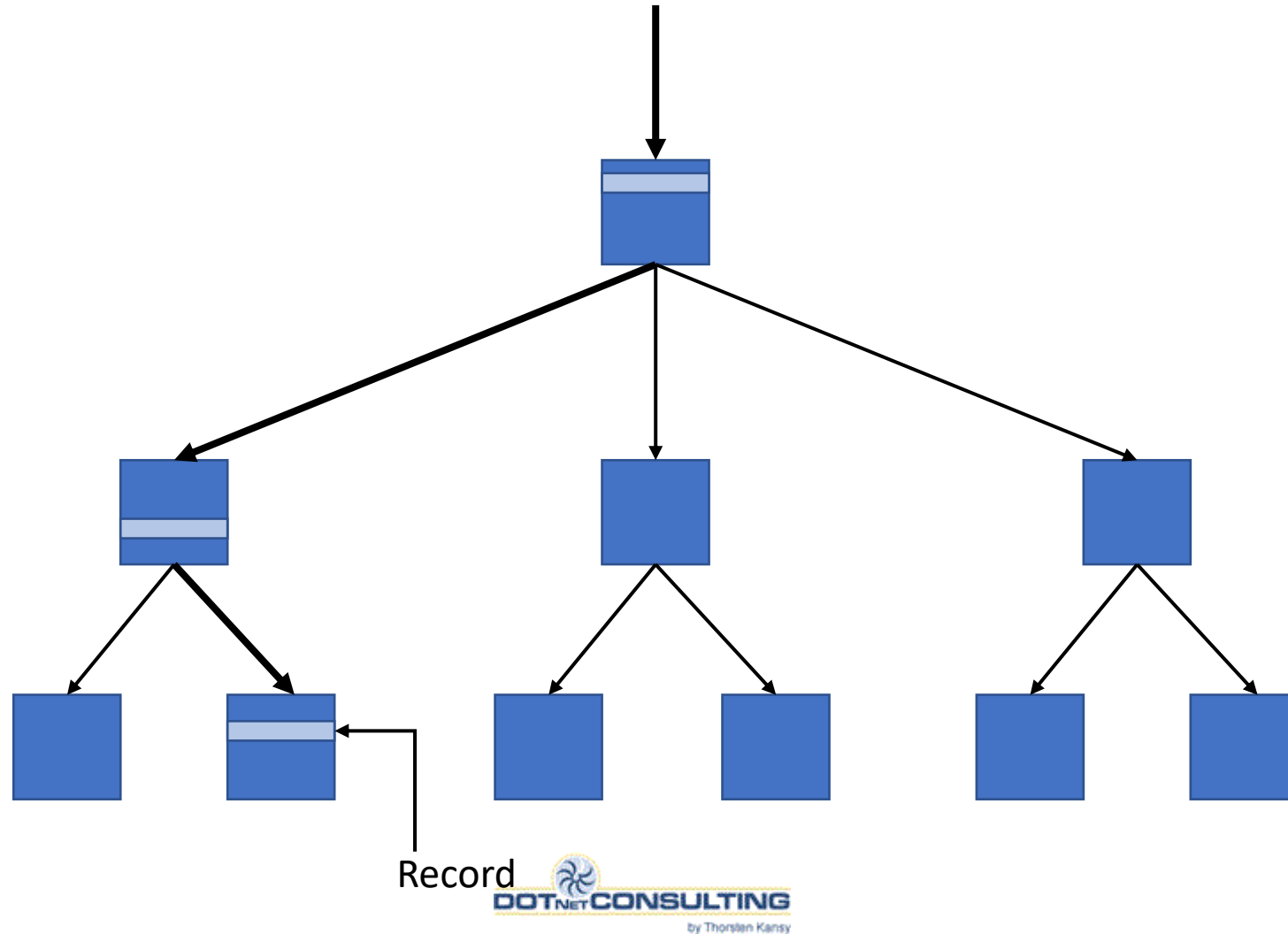
Indexaufbau



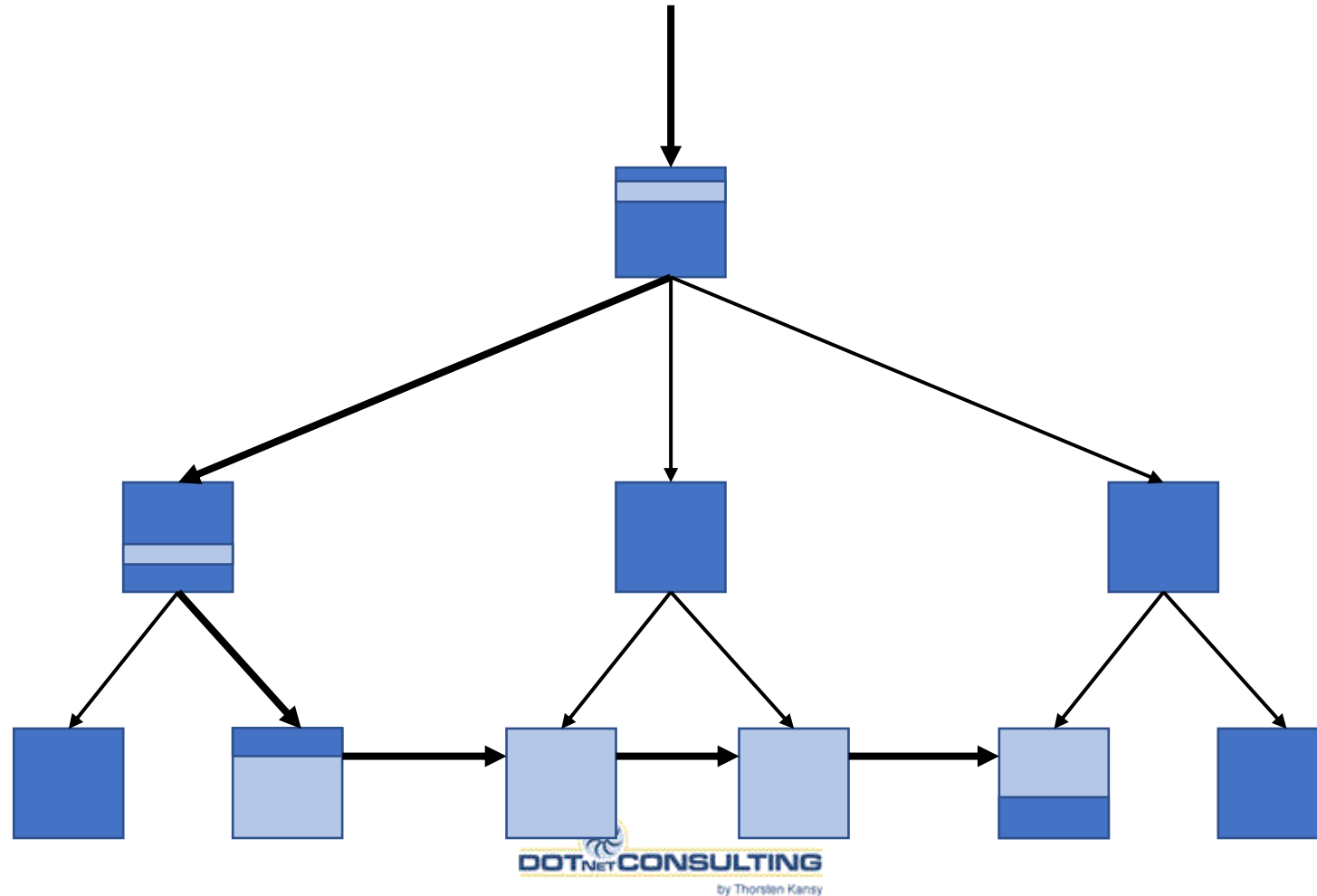
Index Verwendung

- Singleton lookup
- Range scan lookup
 - Readhead
- Full scan

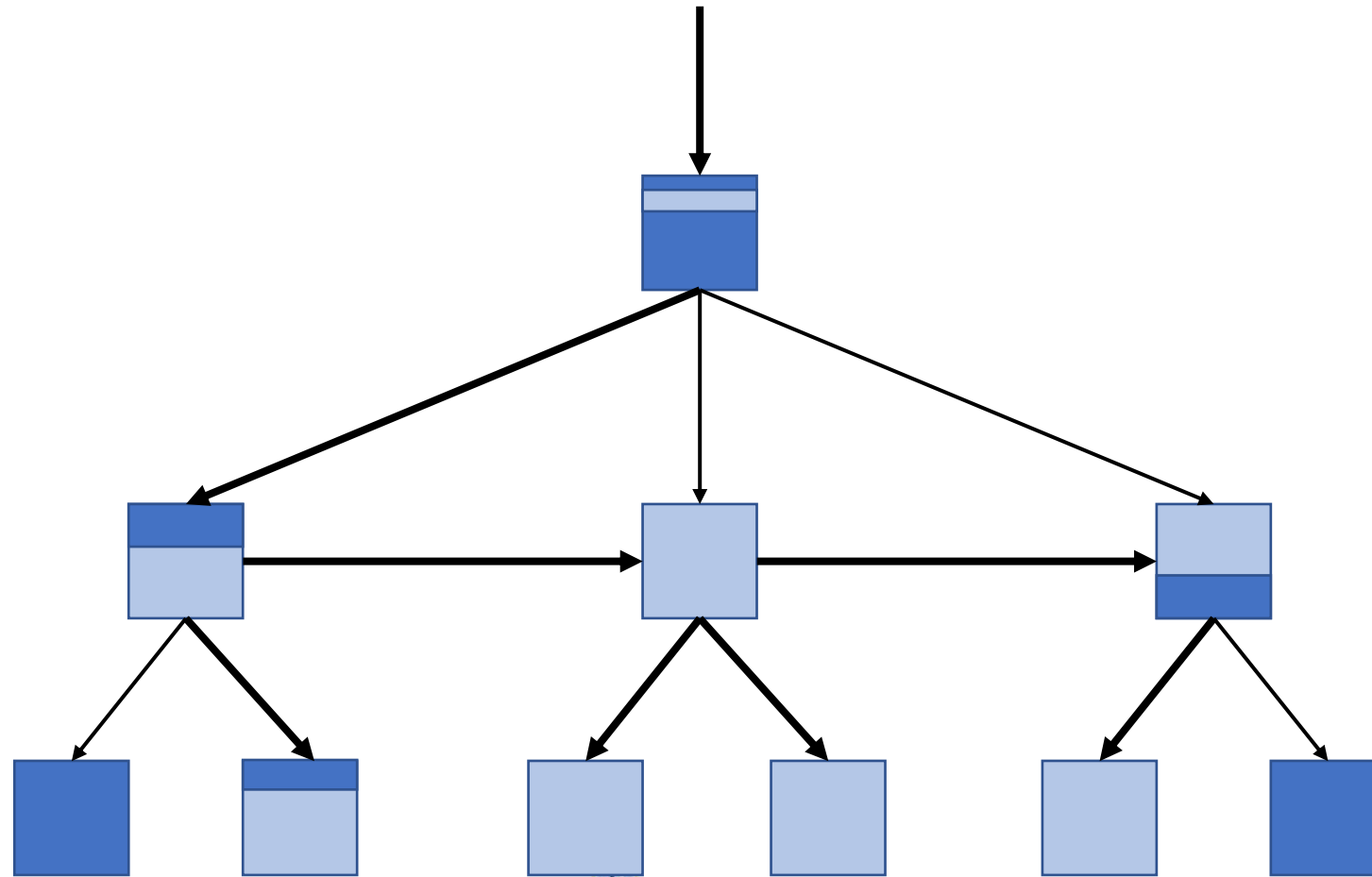
Singelton Lookup



Range scan Lookup



Range-Scan: Readahead



Logical Fragmentation

- Readahead funktioniert nur (gut) bei geringer Fragmentierung
- Weniger bei kleinen Index, mehr bei großen Index

```
SELECT avg_fragmentation_physical_stats FROM  
sys.dm_db_index_physical_stats();
```

- Physikalische Fragmentierung
 - Verteilung der Pages im I/O System
- Logische Fragmentierung
 - Verteilung der Index Records in den Pages

Page Density

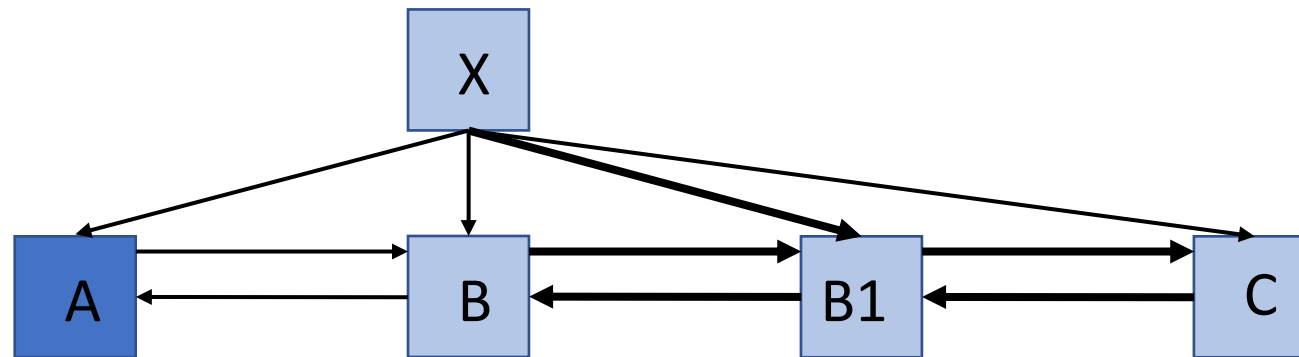
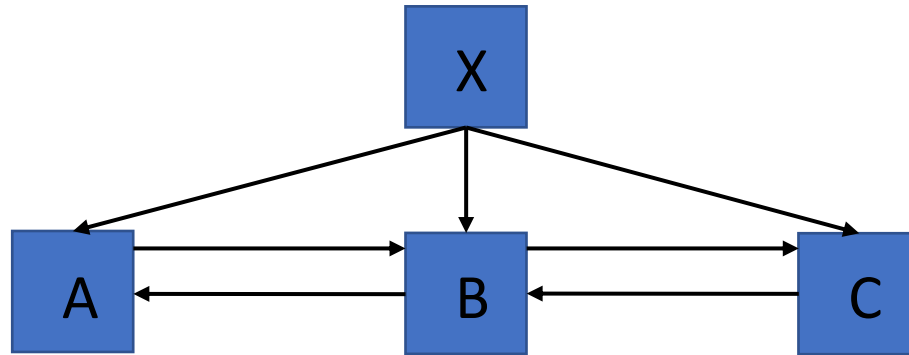
- Niedrige Dichte
 - Verschwendung von I/O Speicher
 - Verstärkte I/O Aktivität
 - Verschwendung von BufferPool RAM Speicher
 - Größere Backups

```
SELECT avg_page_space_used_in_percent FROM  
sys.dm_db_index_physical_stats();
```

Page Split

- Tritt auf weil
 - In einer Page nicht ausreichend Platz ist
 - FILLFACTOR
- Tritt auf wenn
 - Record benötigt mehr Platz als vorhanden
 - neuer Record eingefügt wird
 - Snapshot Isolation: +14 Bytes Versionstore Key

Page split



Page Split erkennen

- Performance Counter:
 - SQLServer: Access Methods > Page Splits/sec
- `SELECT leaf_allocation_count,
nonleaf_allocation_count FROM
dm_db_index_operational_stats(...);`

Effekte der Fragmentierung

- Logical
 - Readahead beeinträchtigt
 - Vergrößertes Transactionsprotokoll => mehr I/O
- Physical
 - Readahead der I/O System beeinträchtigt
- Vergrößertes Transaktionprotokoll

Fillfactor

- Sorgt für Reserve Platz auf Leaf-Level
 - Erstellung/ Rebuild
- Zu große Werte verschwenden Platz
 - SQL liest/ schreibt die komplette Seite!
- Standard per `sp_configure`
- 0 = 100 = keine Reserve
 - Gut: Dataware House
 - Schlecht: OLTP

PAD_INDEX

- Sorgt für Reserve Platz auf nicht Leaf-Level
- Tipp:
 - Gleichen Werte wie FILLFACTOR
 - Per Index
 - Start bei 75%
 - Überwachen
 - Korrigieren

Größe des Index

- Ausnahme: Clustered Index
- Indexierte Spalten
 - Anzahl
 - Datentyp
- Filtered Index
 - `CREATE NONCLUSTERED INDEX <Index> ON <Table> (<Spalten>) WHERE <Prädikat>;`

Wartung

- ALTER INDEX...REBUILD
 - ALTER INDEX...REORGANIZE
 - CREATE INDEX WITH DROP_EXISTING
-
- < 10% oder < 1000 Pages: Nichts!
 - 10-30%: REORGANIZE
 - >30%: REBUILD

Index Strategien

- Wichtig ist ein guter Cluster Index!
- Unique Keys erstellen
- Foreign keys
 - Non clustered indexes
- Mit Bedacht vorgehen!
- Indizes Verwendung überwachen
- Wenn's gut läuft- Finger weg

Fragen?

Links



<http://dotnetconsulting.eu/blog/>



[@Tkansy](https://twitter.com/Tkansy)



tkansy@dotnetconsulting.eu



www.dotnetconsulting.eu