

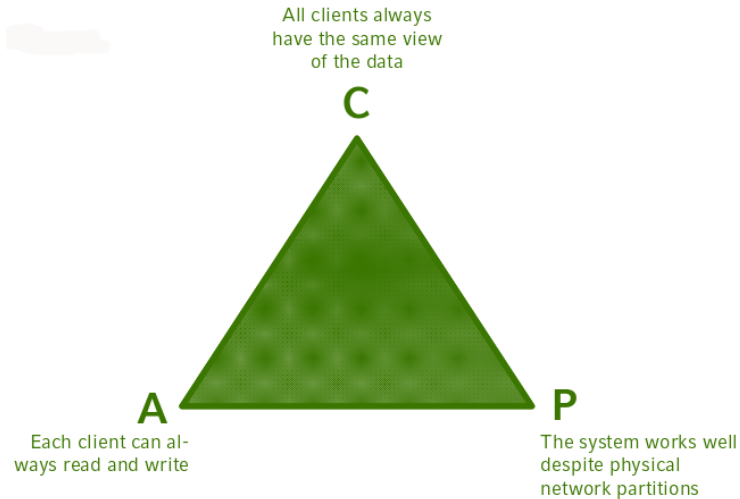
# Big Data Management and Analytics

## Übung 2

# Wiederholung CAP Theorem

- ▶ Consistency: Alle Knoten sehen zur selben Zeit dieselben Daten
- ▶ Availability: Alle Anfragen an das System werden beantwortet
- ▶ Partition Tolerance: Auch bei Verlust von Nachrichten oder Netzknoten arbeitet das System weiter
- ▶ In verteilten Systemen können nur zwei dieser Anforderungen gleichzeitig erfüllt werden

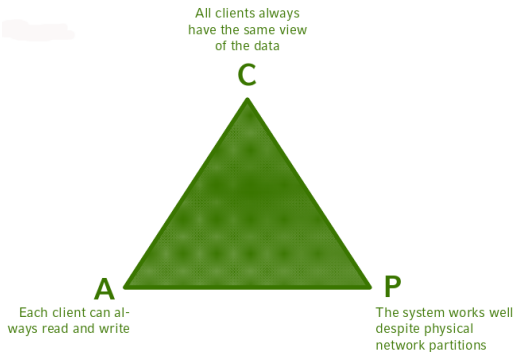
# Wiederholung CAP Theorem



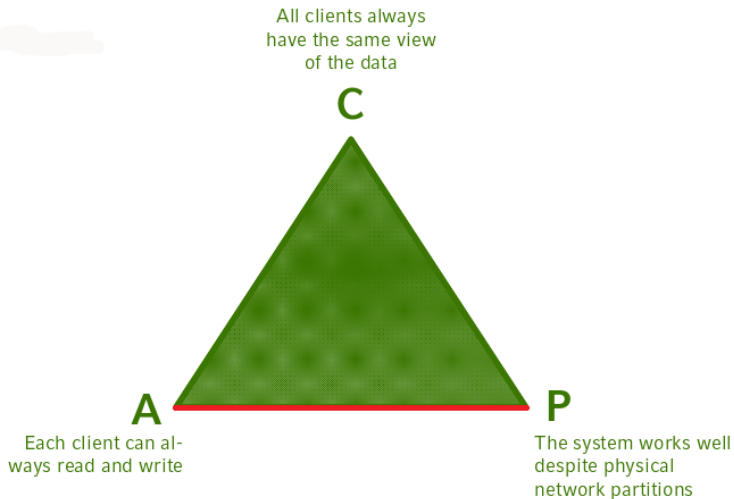
**Aufgabe 1:** Bestimmen Sie für die folgenden Datenbanksystemen, welche der Anforderungen erfüllt werden:

# Aufgabe 1

Bei einem Anbieter eines Cloud-Dienstes werden die Daten auf vielen, nicht unbedingt ausfallsicheren, Knoten verteilt. Diese sollten für dem Kunden nach Möglichkeit immer Verfügbar sein



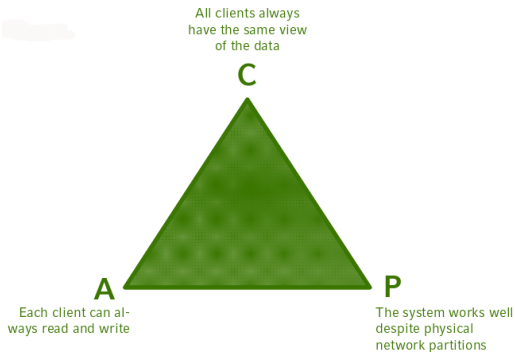
# Aufgabe 1



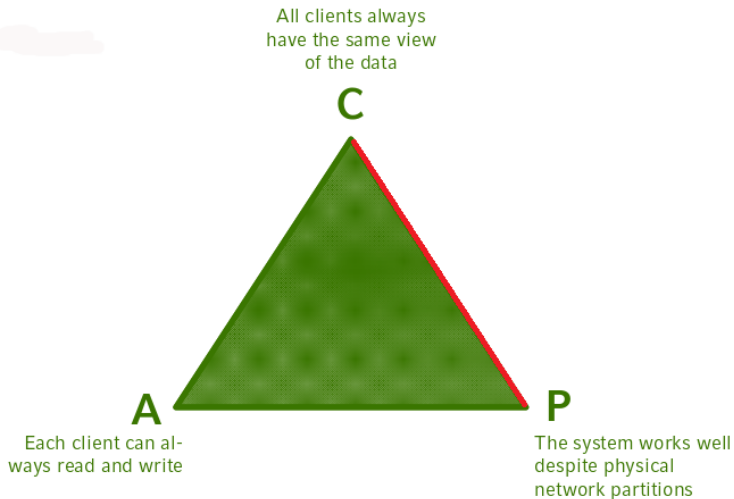
Availability & Partition Tolerance

# Aufgabe 1

Eine größere Bank hat pro Stadt mehrere Geldautomaten. An diesen durchgeführte Transaktionen müssen sorgfältig auf der Datenbank abgespeichert werden, auch wenn der Datenverkehr gestört ist.



# Aufgabe 1

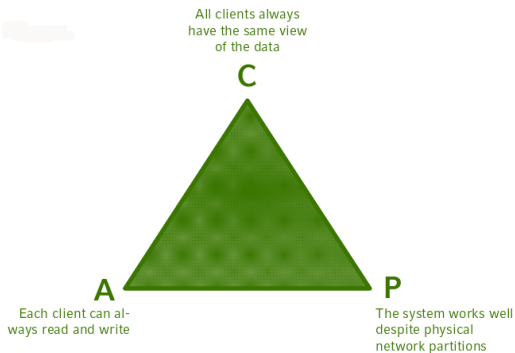


Consistency & Partition Tolerance

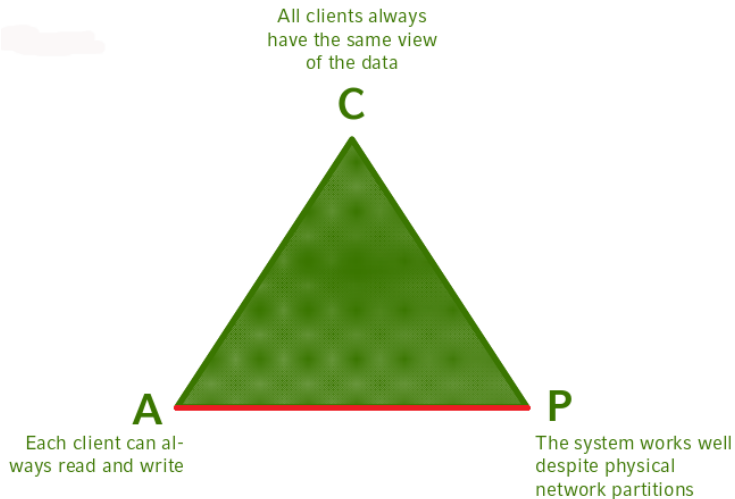


# Aufgabe 1

Bei einem sozialen Netzwerk ist es wichtig, das einmal gepostete Beiträge nicht wieder gelöscht werden. Auch sollte es von seinen Nutzern jederzeit erreichbar sein



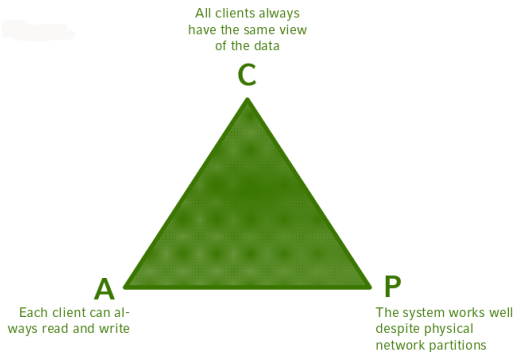
# Aufgabe 1



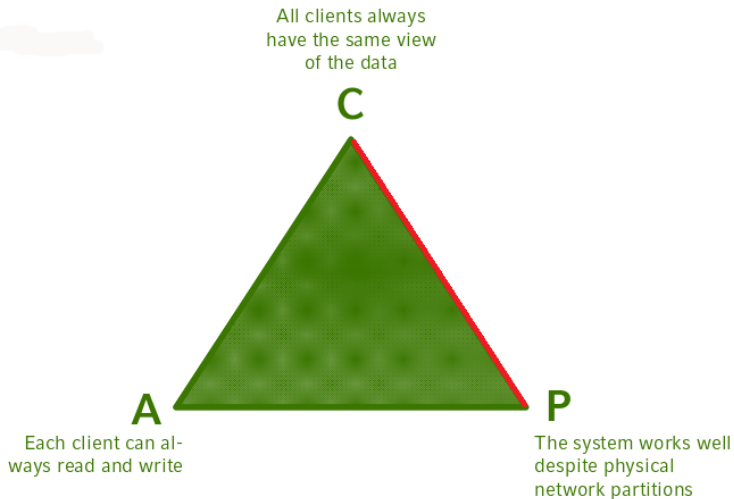
Availability & Partition Tolerance

# Aufgabe 1

Auf der Website eines Hotels können Kunden ein Zimmer online bestellen. Dabei sollen Doppelbuchungen nach Möglichkeit ausgeschlossen werden.



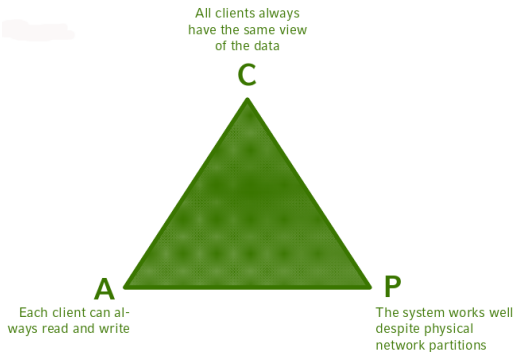
# Aufgabe 1



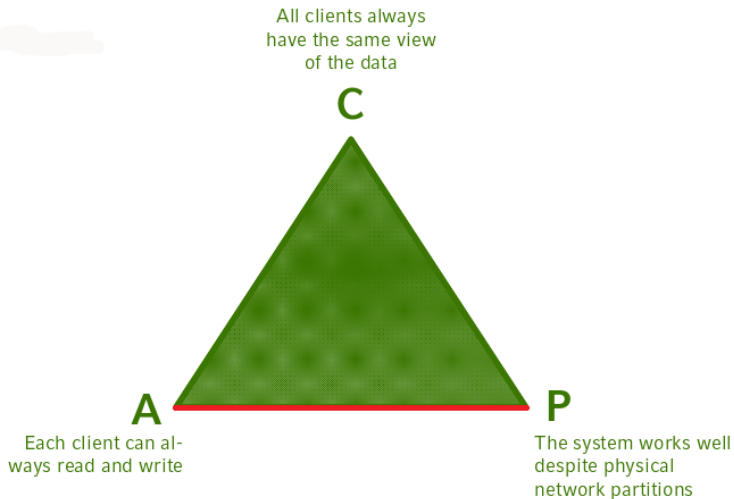
Consistency & Partition Tolerance

# Aufgabe 1

Auf einer Website veröffentlicht eine Nachrichtenagentur regelmäßig Artikel. Diese soll nach Möglichkeit immer aktuell gehalten werden.



# Aufgabe 1



Availability & Partition Tolerance

# MongoDB

Installationsanleitung und Tutorial:  
<https://docs.mongodb.org/manual/>

# MongoDB

Grundlegende Befehle:

```
$mongo
```

```
>use dbName
```

Started mongoshell und wählt Datenbank aus

```
>db.collection.insert(  
{  
<field1:value1>,  
<field2:value2>,...  
}  
)
```

Fügt Wert in Datenbank ein



# MongoDB

Grundlegende Befehle:

```
>db.restaurants.find()  
>db.restaurants.find({  
<field1>: <value1>,  
<field2>: <value2>, ...  
})
```

Gebe alle Einträge einer Collection aus oder suche bestimmte

# MongoDB Replication

- ▶ Daten werden vom primary Knoten auf secondary Knoten kopiert, um Sicherheitskopie zu haben
- ▶ Wenn Primary Knoten ausfällt, wird ein secondary Knoten zum neuen Primary gewählt
- ▶ Daher sollte es immer eine ungerade Anzahl an Knoten geben

# MongoDB Replication

Gebe Replica Set Name:

```
$mongod --replSet "Name"
```

initialisiere Replica Set:

```
$mongo  
>rs.initiate  
>rs.conf()
```

Füge andere Knoten hinzu (mindestens 2):

```
>rs.add("mongodb1.example.net")  
>rs.add("mongodb2.example.net")
```