



# CLOUD-NATIVE

*Unit:*  
**Cloud Computing**

*(2) Crash-Kurs Cloud-Ökonomie*



## Urheberrechtshinweise

Diese Folien werden zum Zwecke einer praktikablen und pragmatischen Nutzbarkeit im Rahmen der **CCo 1.0 Lizenz** bereitgestellt.

Sie dürfen die Inhalte also kopieren, verändern, verbreiten, mit eigenen Inhalten mixen, auch zu kommerziellen Zwecken, und ohne um weitere Erlaubnis bitten zu müssen.

Eine Nennung des Autors ist nicht erforderlich (aber natürlich gern gesehen, wenn problemlos möglich).

Diese Folien sind insb. für die Lehre an Hochschulen konzipiert und machen daher vom **§51 UrhG (Zitate)** Gebrauch.

Die CCo Lizenz überträgt sich nicht auf zitierte Quellen. Hier sind bei der Nutzung natürlich die Bedingungen der entsprechenden Quellen zu beachten.

Die Quellenangaben finden sich auf den entsprechenden Folien.



# KAPITEL 1 + 2

Einleitung, Cloud Computing



## 1 Einleitung

- An wen sich das Buch richtet
- Was dieses Buch behandelt
- Sprachliche und Notationskonventionen
- Ergänzende Materialien

## 2 Cloud Computing

### 2.1 Service Modelle

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

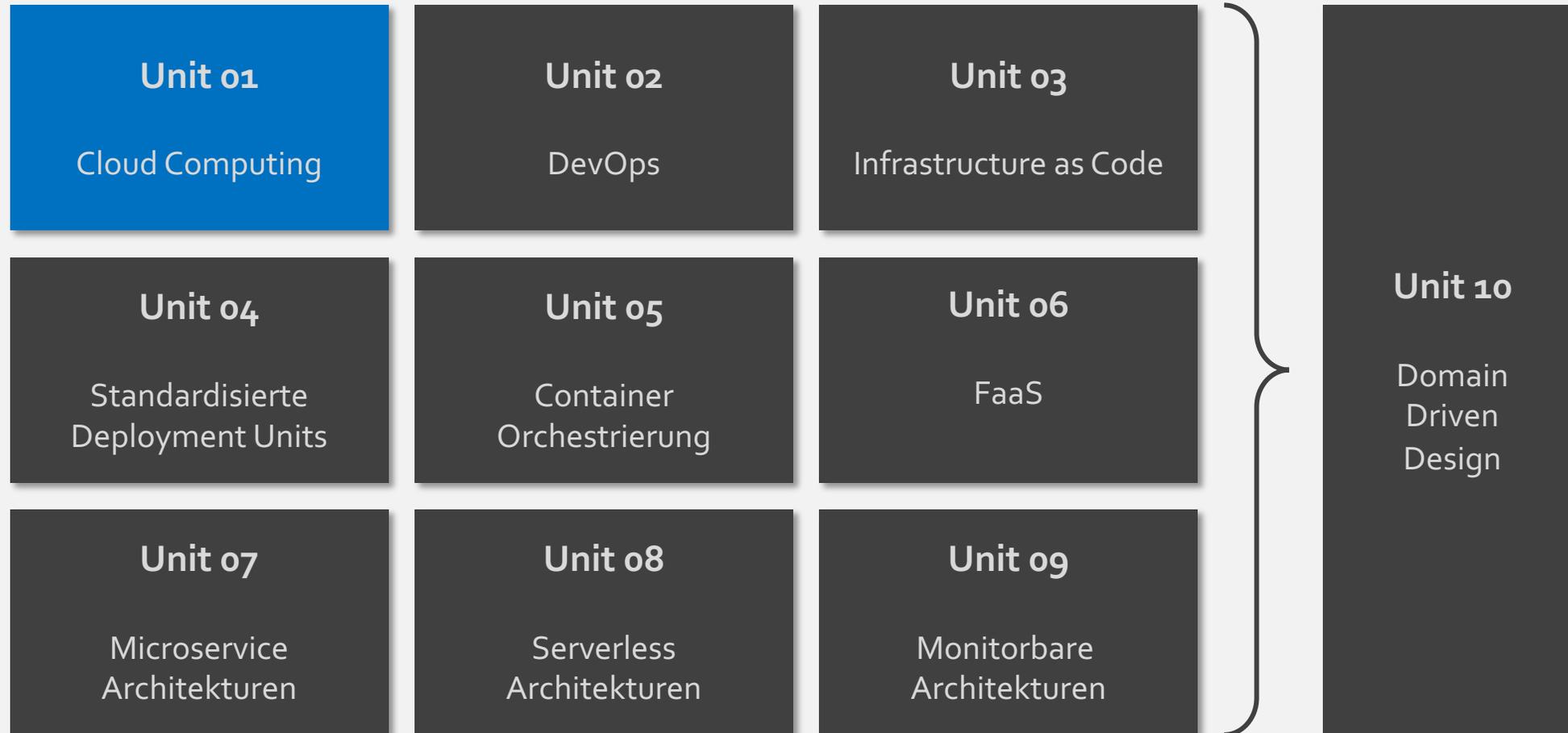
### 2.2 Cloud-Ökonomie

- Eignung von unterschiedlichen Arten von Workloads
- Effekt von Zuteilungsdauer und Ressourcengröße

### 2.3 Entwicklung der letzten Jahre

# INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls

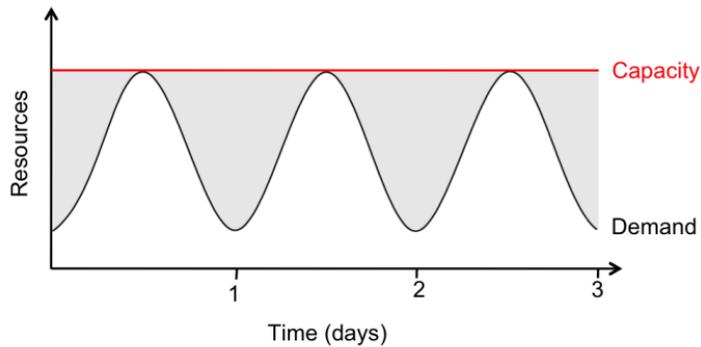


- Cloud Computing
- **Cloud-Ökonomie (Crash-Kurs)**
- Eine kurze Geschichte der Cloud
- Implikationen für die Entwicklung und den Betrieb von Cloud-nativen Anwendungen und Dienste

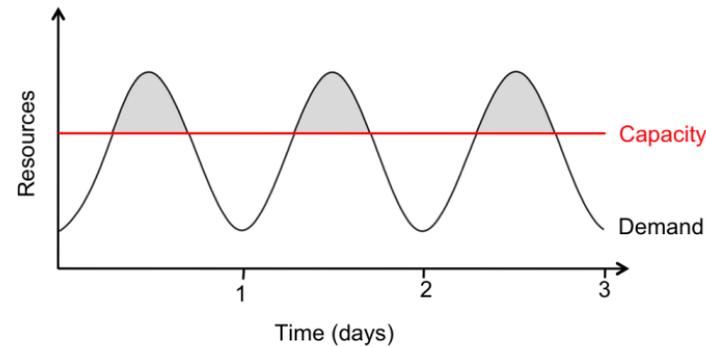


# PAY-AS-YOU-GO

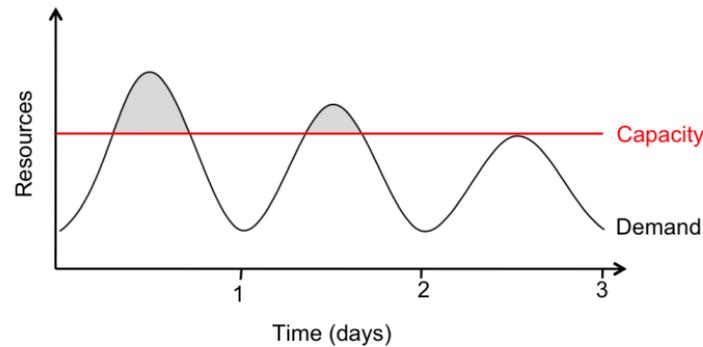
Nur für das Zahlen was man benötigt ...



(a) Provisioning for peak load



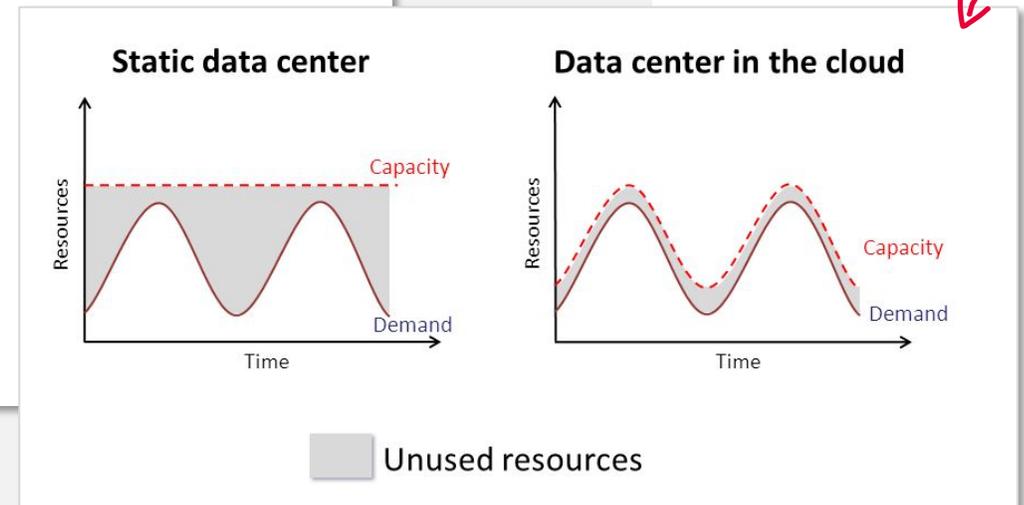
(b) Underprovisioning 1



(c) Underprovisioning 2

Das Dimensionierungsproblem in „klassischen“ On-Premise Rechenzentren.

Cloud Computing ermöglicht Lastkurven enger zu folgen (und Ressourcen zu sparen).

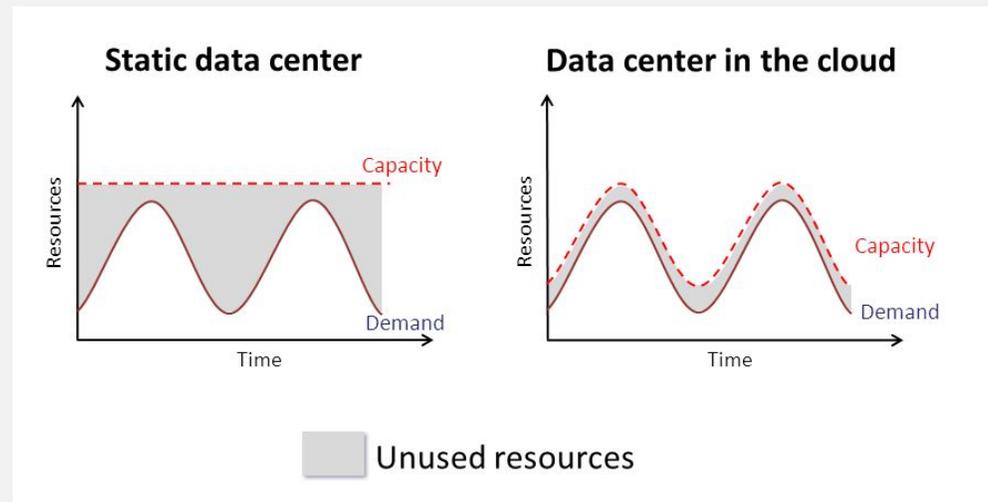


# CRASHKURS IN CLOUD-ÖKONOMIE

*Ist Cloud-Computing immer billiger?*

Cloud-Ressourcen sind vor allem dann wirtschaftlich, wenn Lastschwankungen in einem Anwendungsfall auftreten.

Die Kosten pro Cloud-Ressource können sogar deutlich höher als die In-house Kosten liegen – solange das Verhältnis von Cloud zu In-house Kosten nicht das Verhältnis von Spitzen- zu Durchschnittslast übersteigt.



$$\frac{c}{d} < \frac{p}{a} \Leftrightarrow c < d \frac{p}{a}$$

*d* In-house Aufwand

*c* Cloud-Kosten

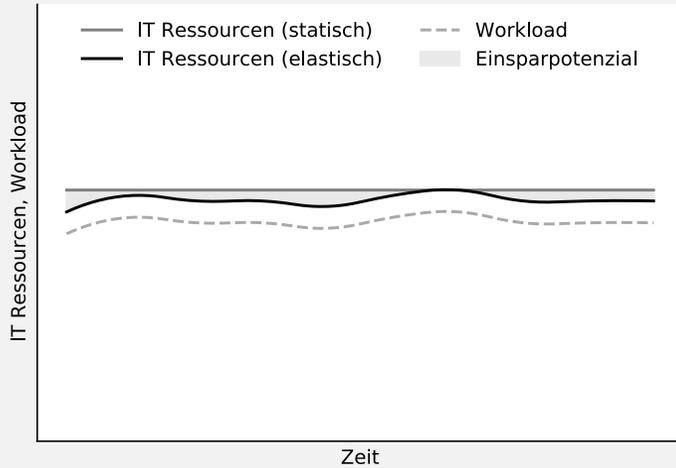
*a* Durchschnittslast  
(average)

*p* Spitzenlast  
(peak)

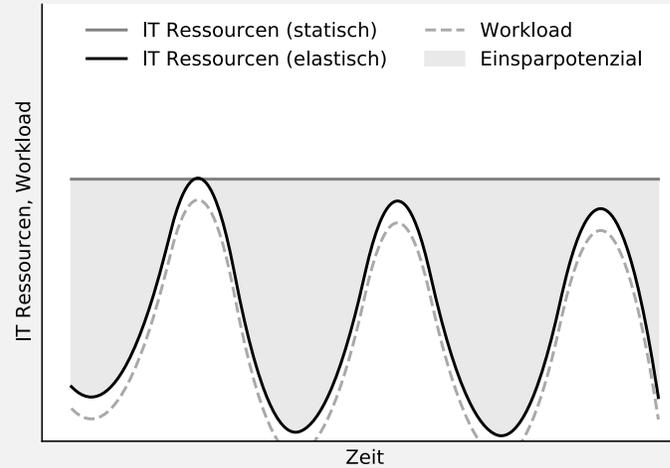
# WORKLOAD KATEGORIEN

Static, Periodic, Unpredictable (Once-in-a-lifetime), Random, Continuously Changing

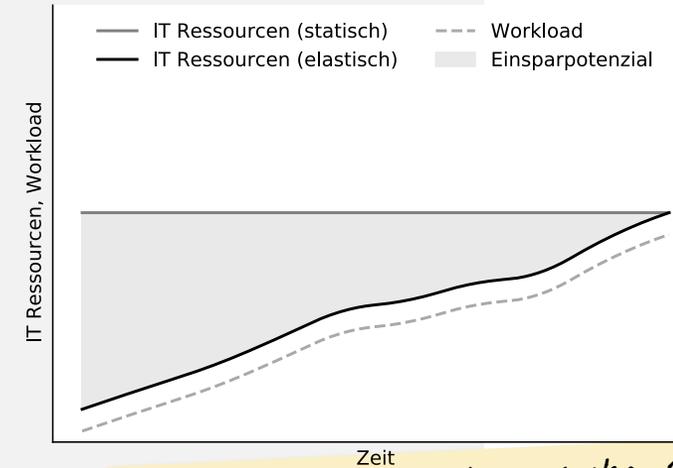
(A) Statischer Workload



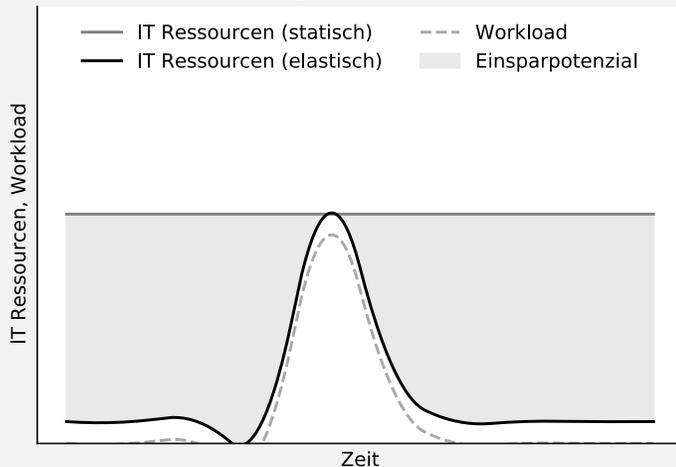
(B) Periodischer Workload



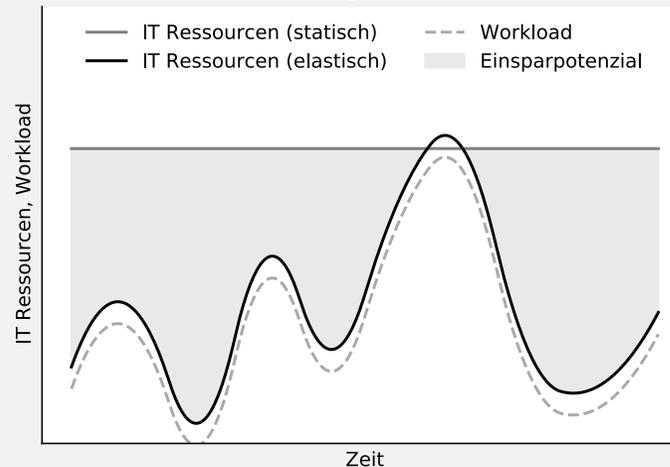
(E) Kontinuierlich wachsender Workload



(C) Einmaliger/seltener Workload



(D) Zufälliger Workload



Workloads des „echten Leben“ setzen sich meist aus diesen Grundtypen zusammen.

Je größer das Peak-to-Average Verhältnis ist, desto wirtschaftlicher sind Cloud-Ansätze

(vorausgesetzt man kann eng genug den Lastkurven folgen).

# PIZZA-AS-A-SERVICE

Ein Beispiel zur Veranschaulichung der Gesetzmäßigkeiten der Cloud-Ökonomie

selbst  
fremd

On-Premise	IaaS	PaaS	SaaS
Esstisch	Esstisch	Esstisch	Esstisch
Getränke	Getränke	Getränke	Getränke
Ofen	Ofen	Ofen	Ofen
Belag	Belag	Belag	Belag
Tomatensauce	Tomatensauce	Tomatensauce	Tomatensauce
Pizzateig	Pizzateig	Pizzateig	Pizzateig
Zutaten	Zutaten	Zutaten	Zutaten
<i>Pizza di Mama</i>	<i>Kaufen &amp; Backen</i>	<i>Pizza Service</i>	<i>Pizzeria</i>

*Wir fragen uns nun, ob es Pizzakonsum-Gewohnheiten geben könnte, die Service-Modelle wirtschaftlich vorteilhafter erscheinen lassen könnten.*

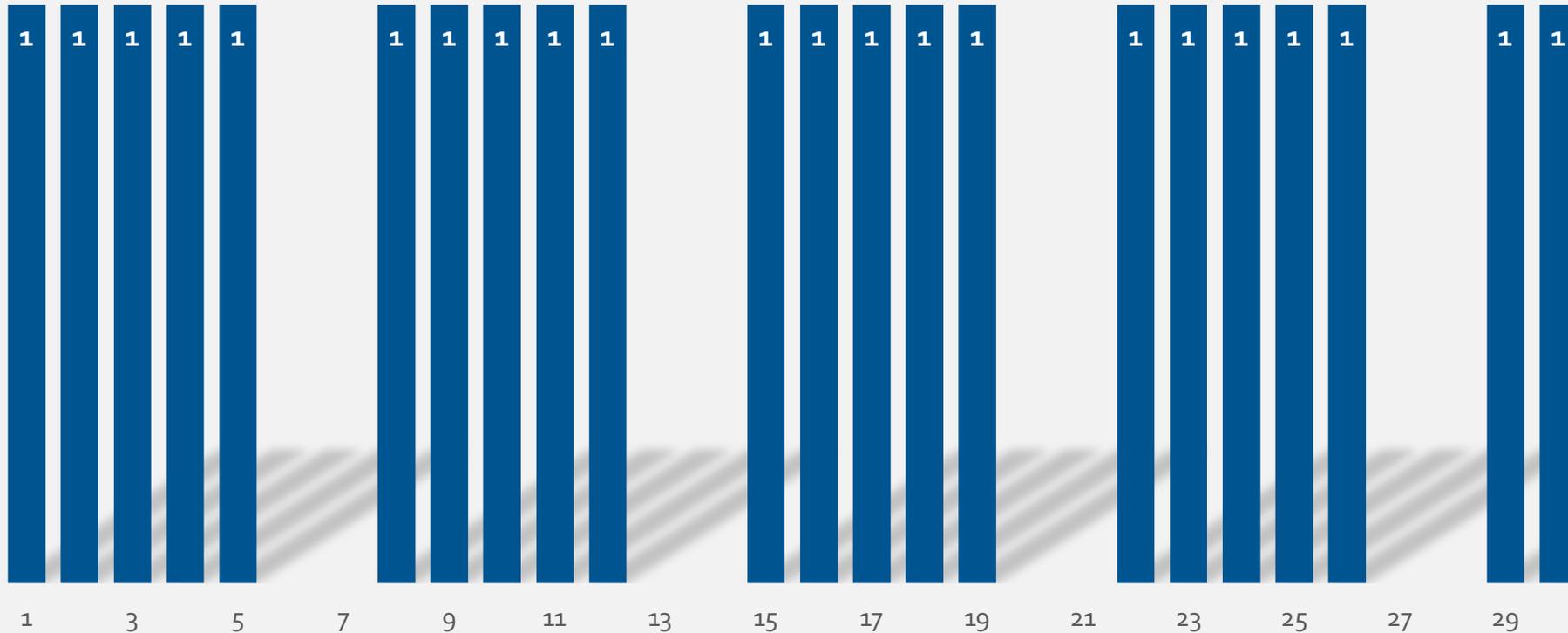
$$\frac{c}{d} < \frac{p}{a} \Leftrightarrow c < d \frac{p}{a}$$

# STATISCHE WORKLOADS

Der regelmäßige „Fastfood“ Workload



Der Cloud-Provider kann bis zu **30%** teurer sein als Ihre selbstgemachte Pizza.



Sie kaufen sich an jedem Werktag zur Mittagszeit eine Pizza am Stand gegenüber vor Ihrer Arbeitsstelle.

An Wochenenden tun Sie das natürlich nicht.

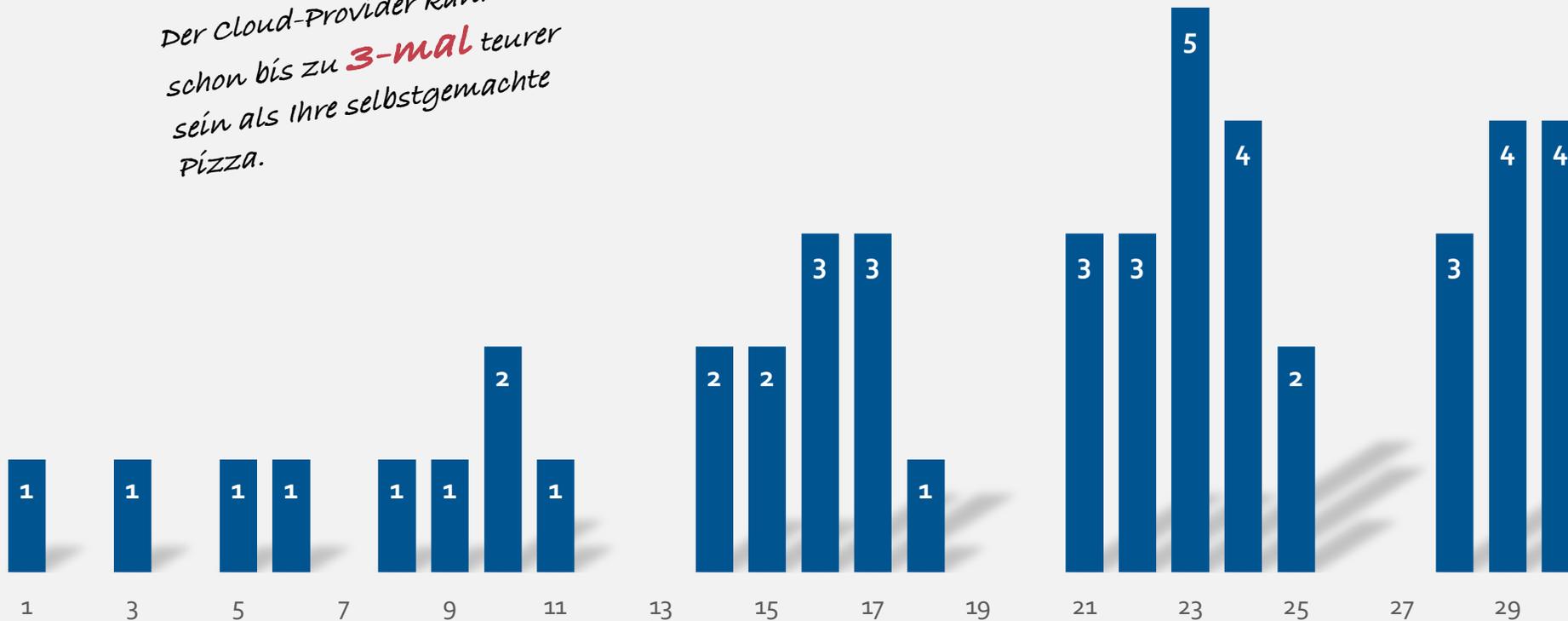
$$p = 1$$
$$a = 22/30$$

$$p/a \approx 1.3$$

# CONTINUOUSLY-CHANGING WORKLOADS

## Der Trendsetter-Workload

Der Cloud-Provider kann nun schon bis zu **3-mal** teurer sein als Ihre selbstgemachte Pizza.



Sie bringen Ihren Kollegen immer was vom Pizzawagen mit.

Das spricht sich rum, und Woche für Woche müssen Sie mehr Pizza besorgen.

An Wochenenden arbeitet natürlich niemand.

$$p = 5$$
$$a = 48/30$$

$$p/a \approx 3.1$$

# PERIODISCHE WORKLOADS

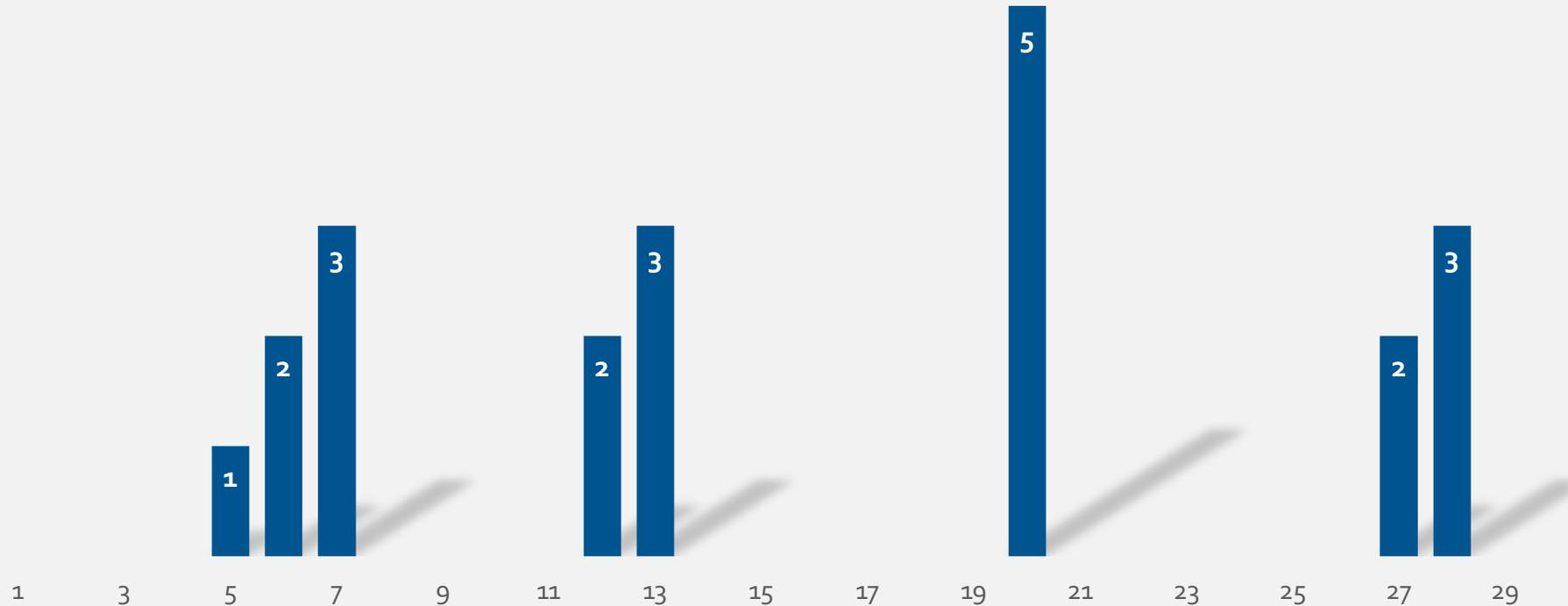
## Der DVD-Abend/Party-Workload

Ihr Bedarf wird seltener und der Cloud-Provider kann nun schon bis zu **7-mal** teurer sein als Ihre selbstgemachte Pizza.



Sie machen mit Familie und Freunden an Wochenenden DVD-Abende und reichen dazu Pizza.

Unter der Woche haben Sie dazu natürlich keine Zeit.

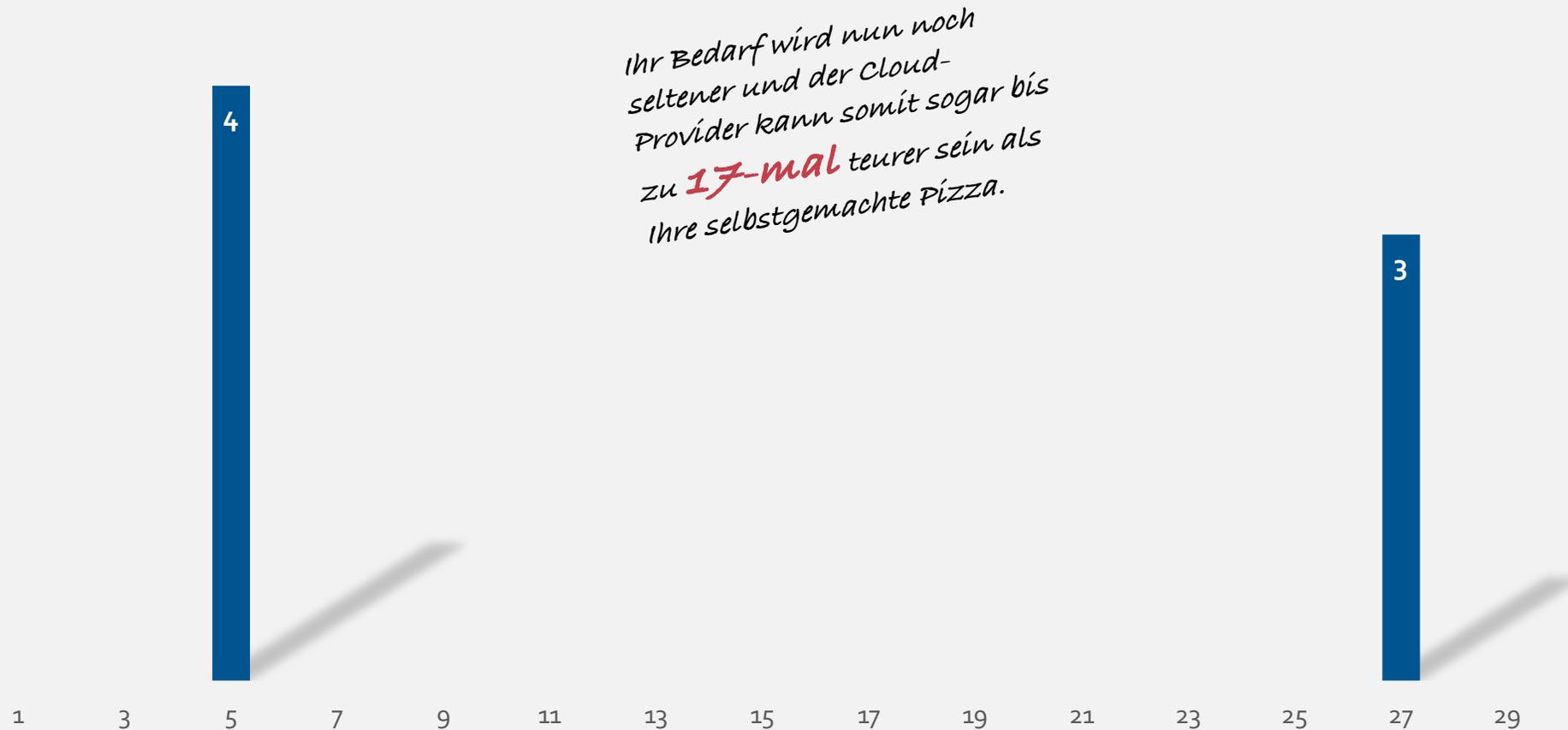


$$p = 5$$
$$a = 21/30$$

$$p/a \approx 7.1$$

# UNVORHERSEHBARE/SELTENE WORKLOADS

## Der Pizzeria-Workload



*Sie laden Ihre Familie an Wochenenden ab und an in eine Pizzeria ein.*

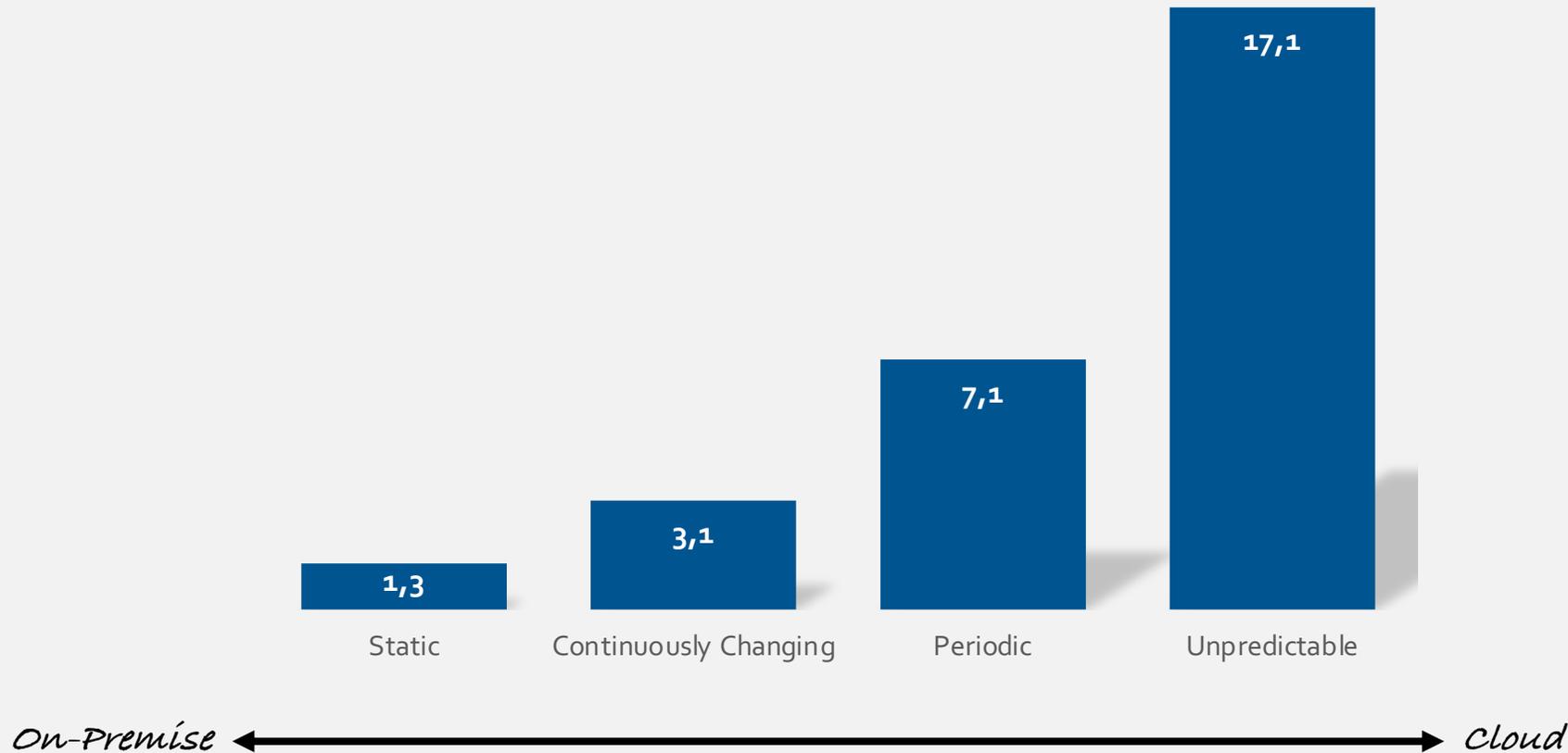
*Unter der Woche haben Sie dazu natürlich keine Zeit.*

$$p = 4$$
$$a = 7/30$$

$$p/a \approx 17.1$$

# KOSTENVORTEILE

Entstehen also in erster Linie durch die Art des Workloads!



*Kostenvorteile entstehen also in aller Regel durch den Workload und erst in zweiter Linie durch die Kostenstruktur des Dienstes.*

**Fun Fact:**  
*AWS hat ca. 50% Marktanteil im Cloud Computing verlangt aber häufig bis zu 10% mehr pro Einheit für Services als bspw. Google.*

**Warum ist das so?**

*Vielleicht weil bei diesen Kostenvorteilen 10% Preisunterschiede kaum noch messbar sind?*

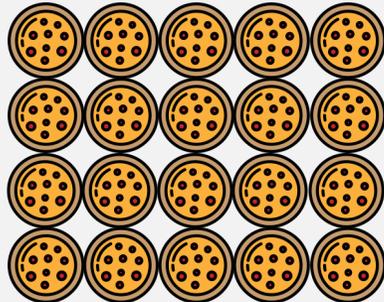
# FUN FACT

Es kostet übrigens dasselbe ...

*Diese Tatsache nennt sich  
Kostenassoziativität.*

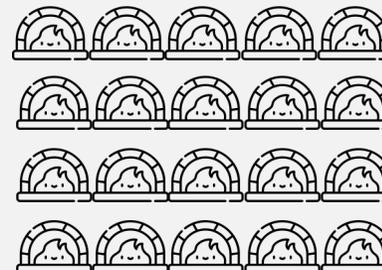
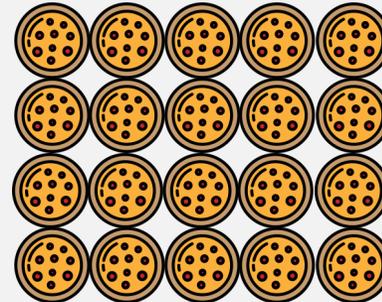
*Cloud-Ökonomie  
kreativ nutzen*

1 Ofen für 20 x 15 min zu mieten



*Dauer:  
5 Stunden*

20 Öfen für 15 min zu mieten



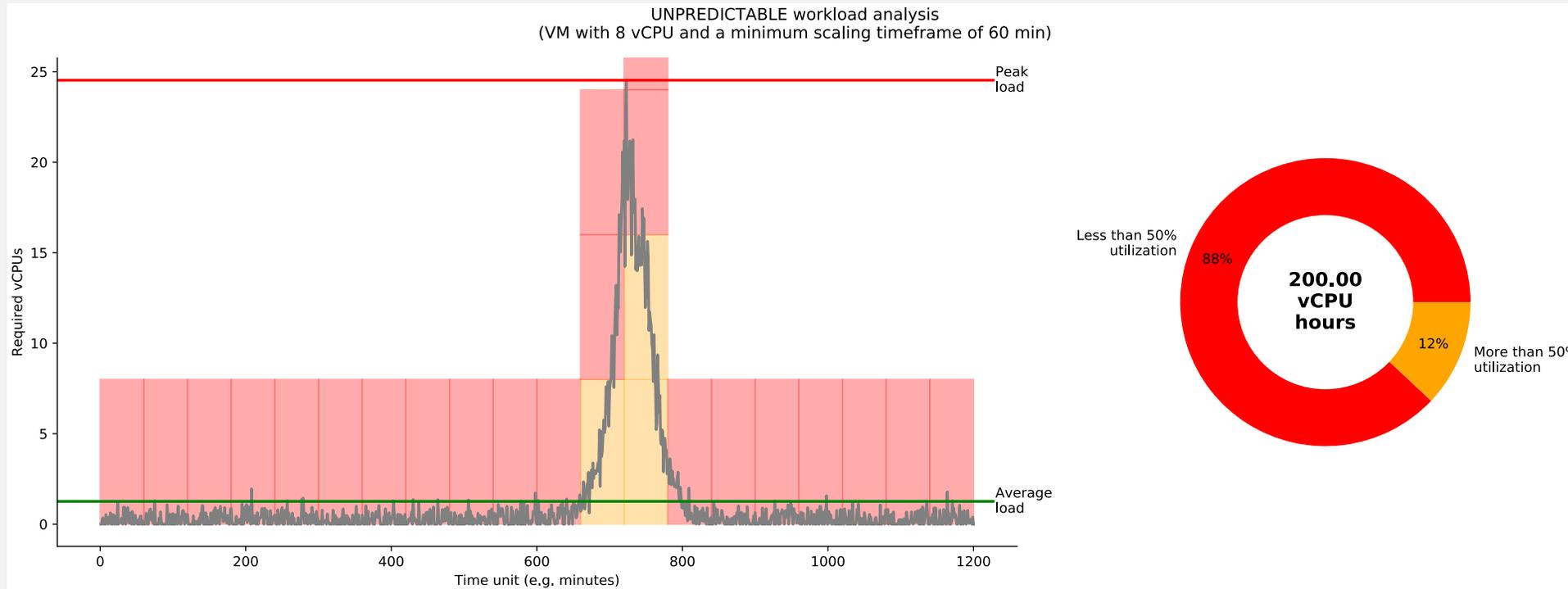
*Dauer:  
15min*

*Bei welchem  
Lieferdienst würden  
Sie 20 Pizzen  
bestellen?*

- Bei dem der Sie in 5 Stunden beliefert und bei dem 19 Pizzen kalt sind?*
- Bei dem, der Ihnen nach 15 Minuten 20 warme Pizzen liefern kann?*
- Wieviel Aufpreis wäre Ihnen das wert?*
- Wieviel Mehraufwand kostet das den Lieferdienst?*
- Wie häufig brauchen Sie als Lieferdienst wohl 20 Öfen gleichzeitig?*

# DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Am Beispiel der Unpredictable Workload Kategorie



*Ausgangssituation:  
Nutzung von  
großen virtuellen  
Maschinen für  
„lange“ Zeiträume*

*VM: 8 vCPU  
Dauer: 60 min*

*Resultierende  
Ressourcen-  
anforderung:  
200 CPU-Stunden*

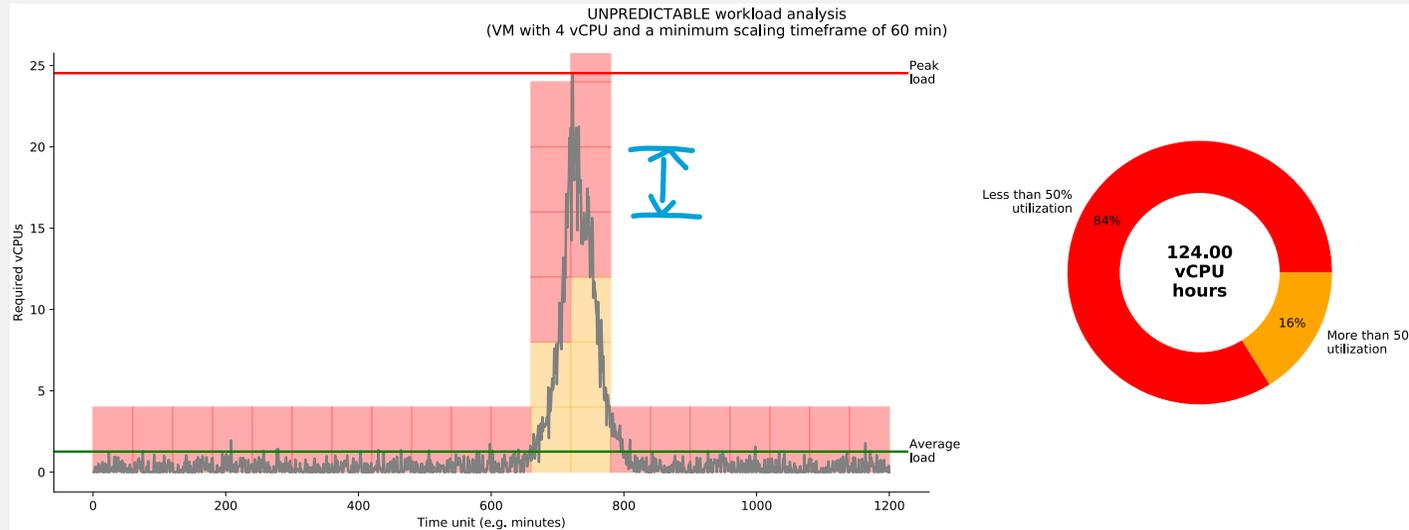
Wir sehen in diesem Modell recht ineffizient genutzte VMs. Aber ungenutzte VMs kosten dasselbe wie genutzte VMs. Die Ursache liegt letztlich in zu „großen Kästen“.

An welchen Stellschrauben kann man also drehen, um die Ressourcennutzung effizienter zu machen (also die Kästen kleiner zu machen)?

*Das ist im Wesentlichen  
der Kostentreiber!!!*

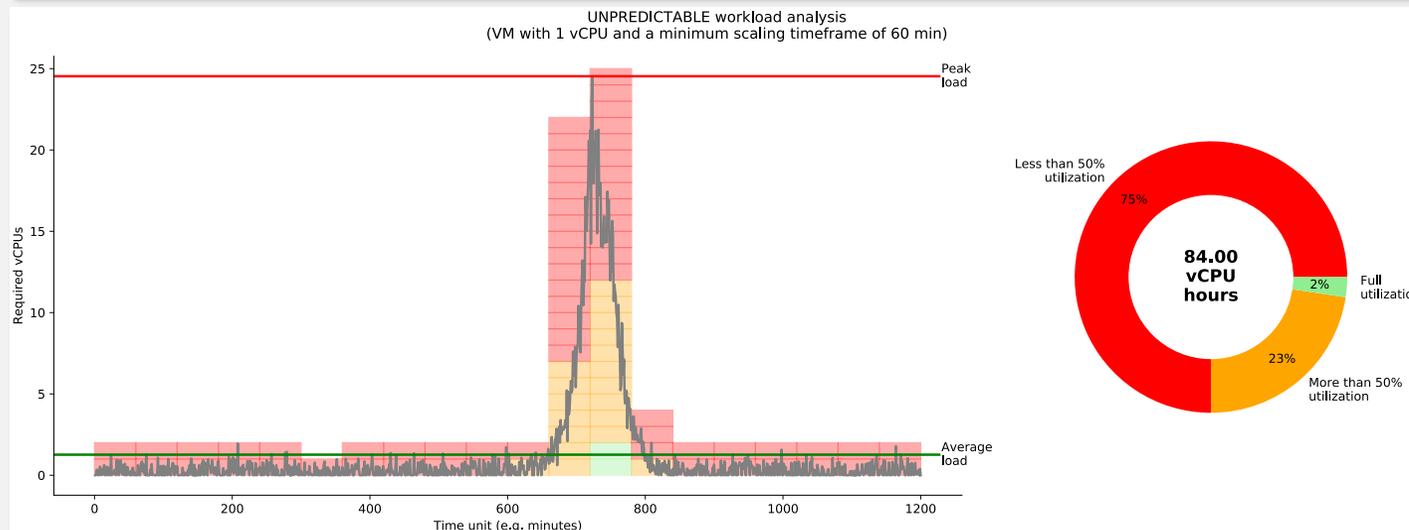
# DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Was passiert wenn wir die VM Größe reduzieren?



VM: 4 vCPU  
Dauer: 60 min  
=> 124 CPU-Stunden

Mit kleineren VMs benötigt man zwar mehr VMs, aber fordert dennoch weniger CPU-Stunden insgesamt an.



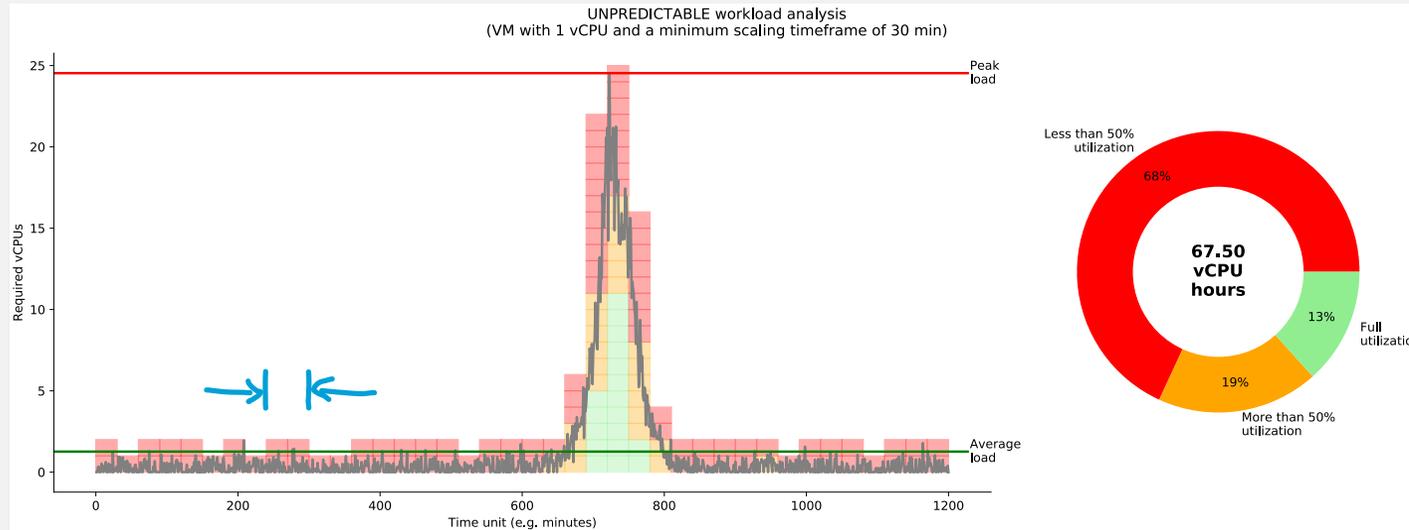
VM: 1 vCPU  
Dauer: 60 min  
=> 84 CPU-Stunden

Mit kleineren VMs kann man Lastkurven also „enger“ folgen und verschwendet weniger ungenutzte Ressourcen.

Stellschraube 1:  
VM-Größe

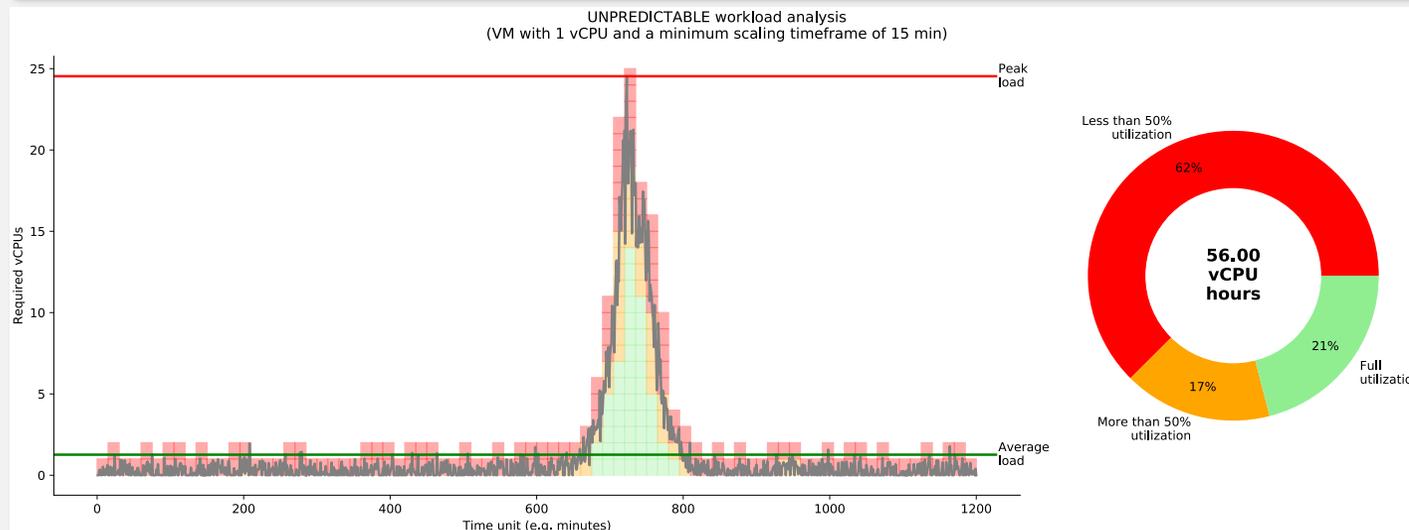
# DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Was passiert wenn wir die Zuteilungsdauer reduzieren?



VM: 1 vCPU  
Dauer: 30 min  
=> 67.5 CPU-Stunden

Mit kürzeren Zuteilungsdauern kann man Lastkurven also „schneller“ folgen und verschwendet kürzer ungenutzte Ressourcen.



VM: 1 vCPU  
Dauer: 15 min  
=> 56 CPU-Stunden

Alle Cloud Provider haben mit einer Stunden-basierten Abrechnung begonnen, und haben dann auf 30min, 15min, 5min, 1min Abrechnungsintervalle umgestellt (einige auch bereits auf Sekunden-basis).

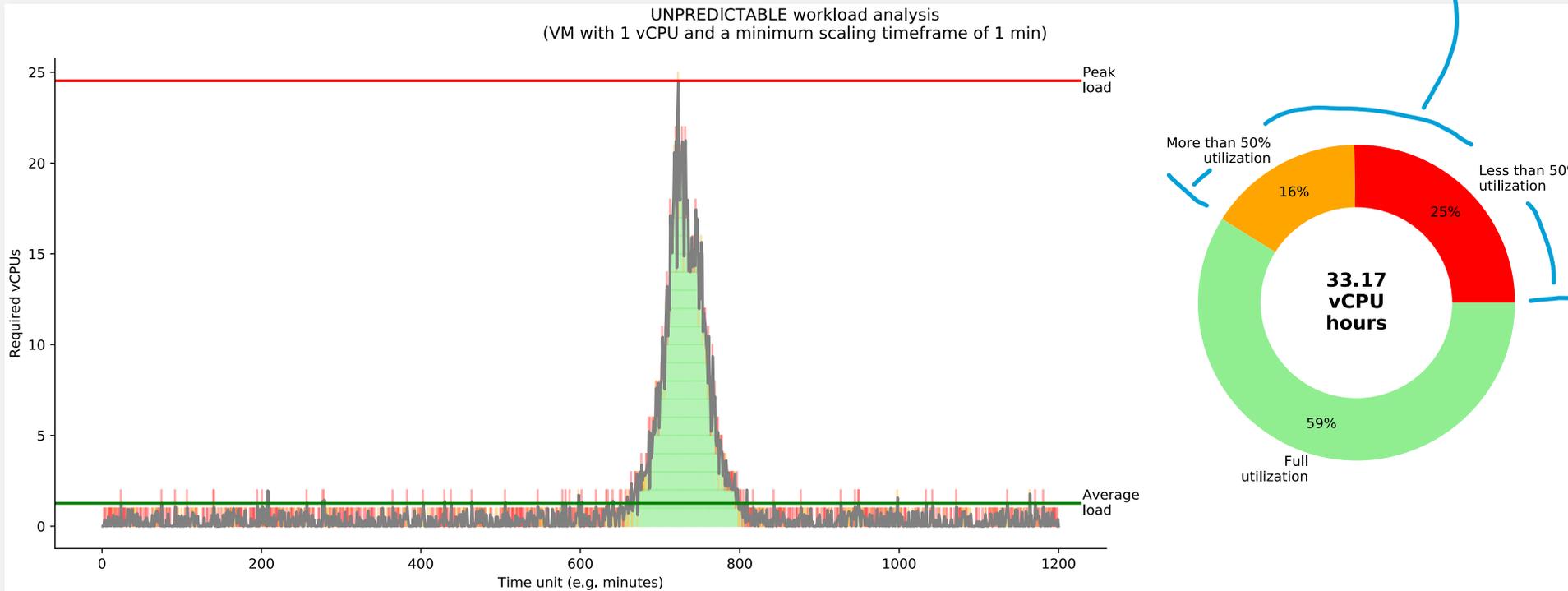
Stellschraube 2:  
**Zeit**  
(Abrechnungsintervalle)

Den maximalen „Effekt“ den Sie damit erzielen können, sehen Sie auf der linken Seite!

# DER EFFEKT FREINGRANULARER ZUTEILUNGEN

Wo liegen die Grenzen dieses IaaS-fokussierten Ansatzes?

Hier scheint noch  
weiteres Optimierungspotenzial zu liegen.



VM: 1 vCPU  
Dauer: 1 min

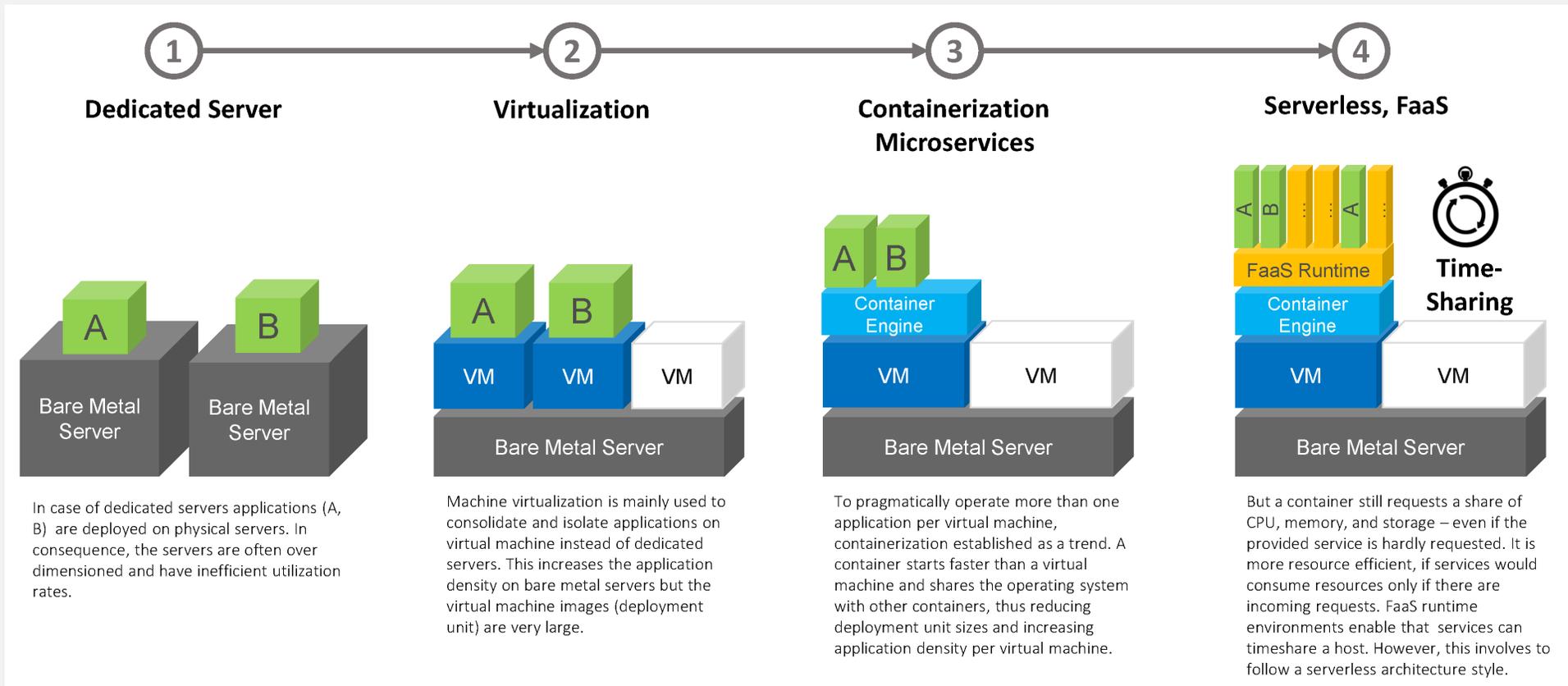
⇒ 33.17 CPU-Stunden

um in diesen Bereich vorzudringen müssen wir allerdings noch schneller und noch feingranularer skalieren können.

- Container (Sub Core Scale)
- Faas (Sub Second Scale)

# HISTORISCHE ENTWICKLUNG

Eine kurze Geschichte der Cloud



*Im verlaufe der Zeit sind die Größen von Deployment Units geschrumpft!*

*VM -> Container  
-> Function*

*Parallel dazu haben sich Servis-orientierte Architekturen (weiter)entwickelt.*

*Microservices + Serverless Architekturen*

- Cloud Computing
- Cloud-Ökonomie (Crash-Kurs)
- **Eine kurze Geschichte der Cloud**
- Implikationen für die Entwicklung und den Betrieb von Cloud-nativen Anwendungen und Dienste



# KONTAKT

*Disclaimer*

**Nane Kratzke**

📞 +49 451 300-5549

✉ nane.kratzke@th-luebeck.de

🔗 [kratzke.mylab.th-luebeck.de](https://kratzke.mylab.th-luebeck.de)

