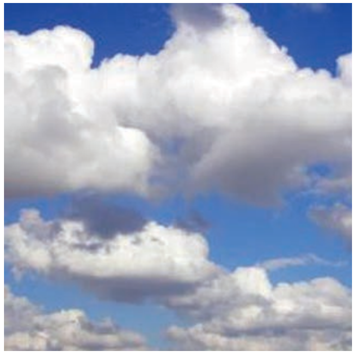
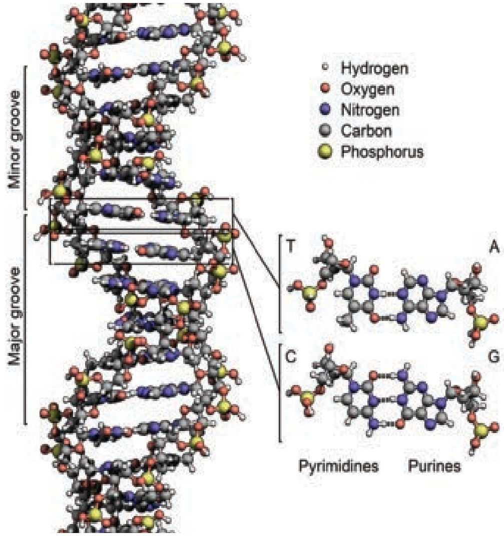
Reducing Energy Consumption and Improving Bioinformatics with Clouds



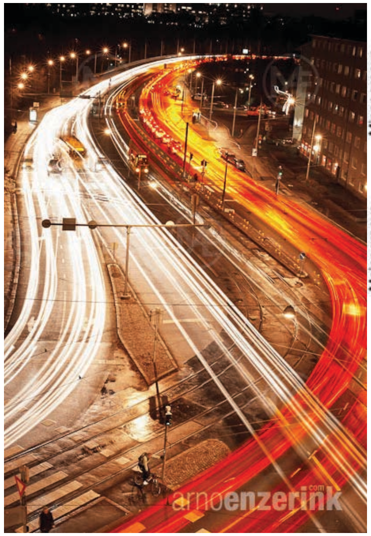


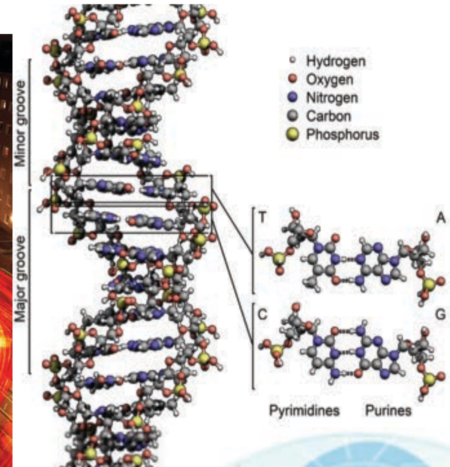
Ivona Brandić

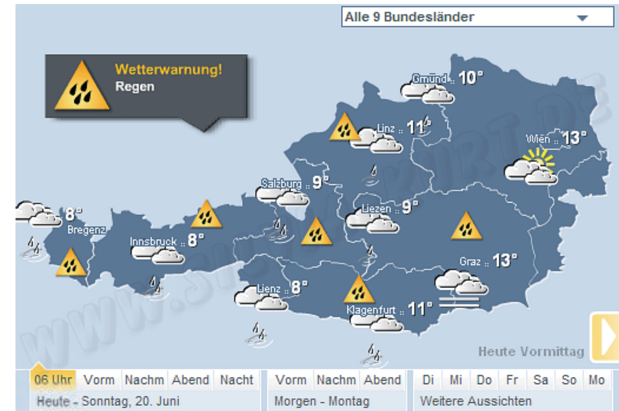
Distributed Systems Group
Institute of Information Systems, Vienna University of Technology
Austria

ivona@infosys.tuwien.ac.at

Current challenge of our society: BIG DATA!





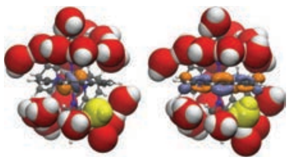


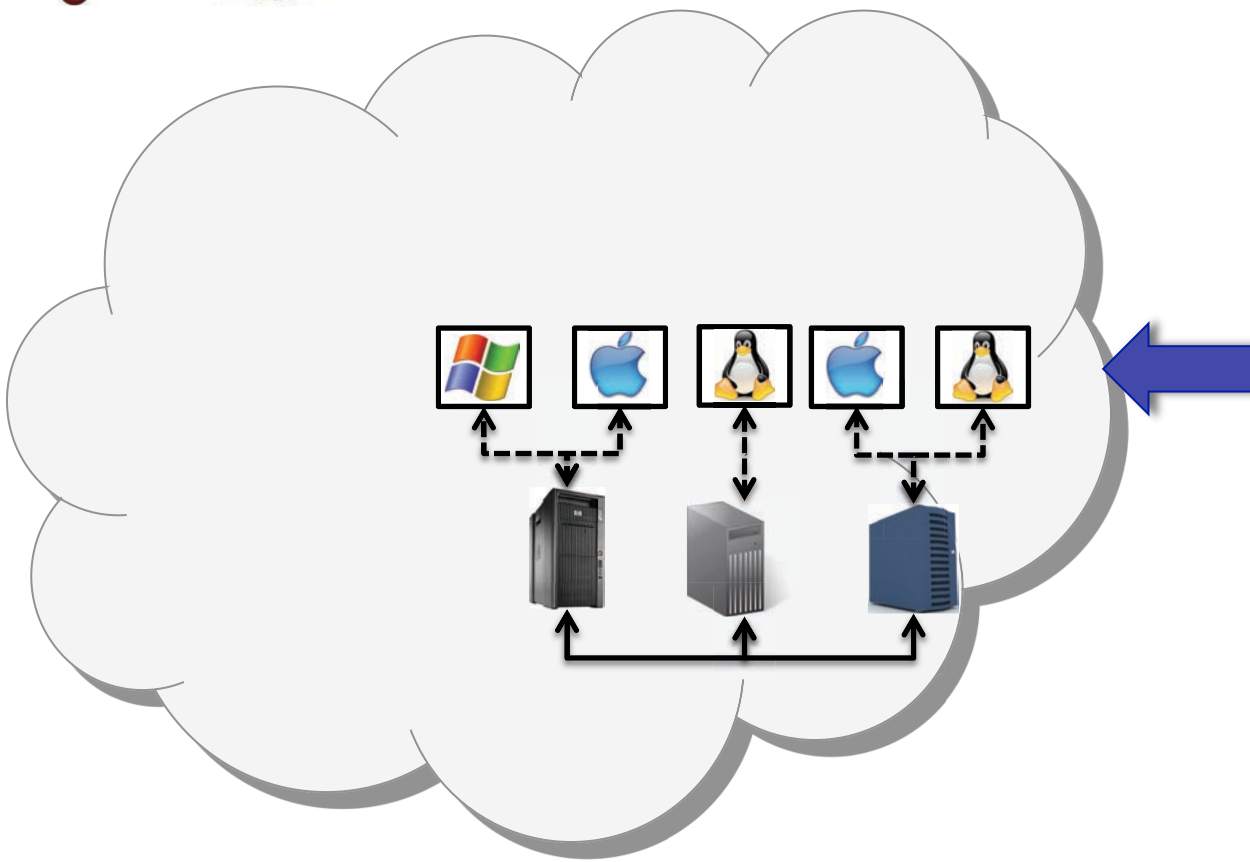
Big Data vs. Energy Efficiency

- ICT consumes **10.5% of energy in Germany** and rising, (Deutscher Bundestag, 2010)
- ICT **produces 2% of worldwide CO_2 emissions and rising -** equivalent to aviation industry, (Gartner, 2007)

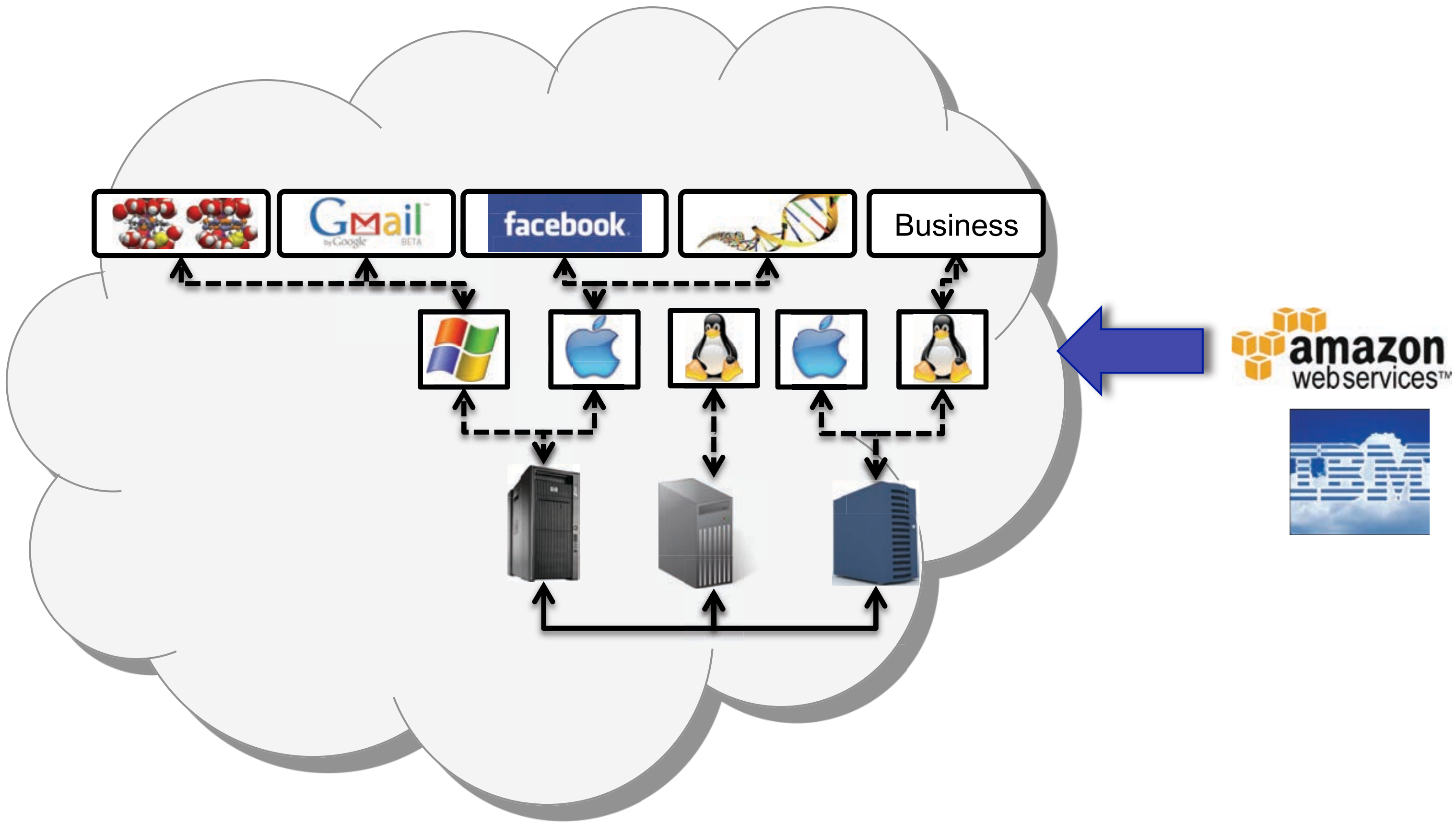
ARE THE CLOUDS CANDIDATE (ENERGY EFFICIENT) SOLUTION FOR BIG DATA PROBLEMS?

Cloud Computing and Energy Efficiency





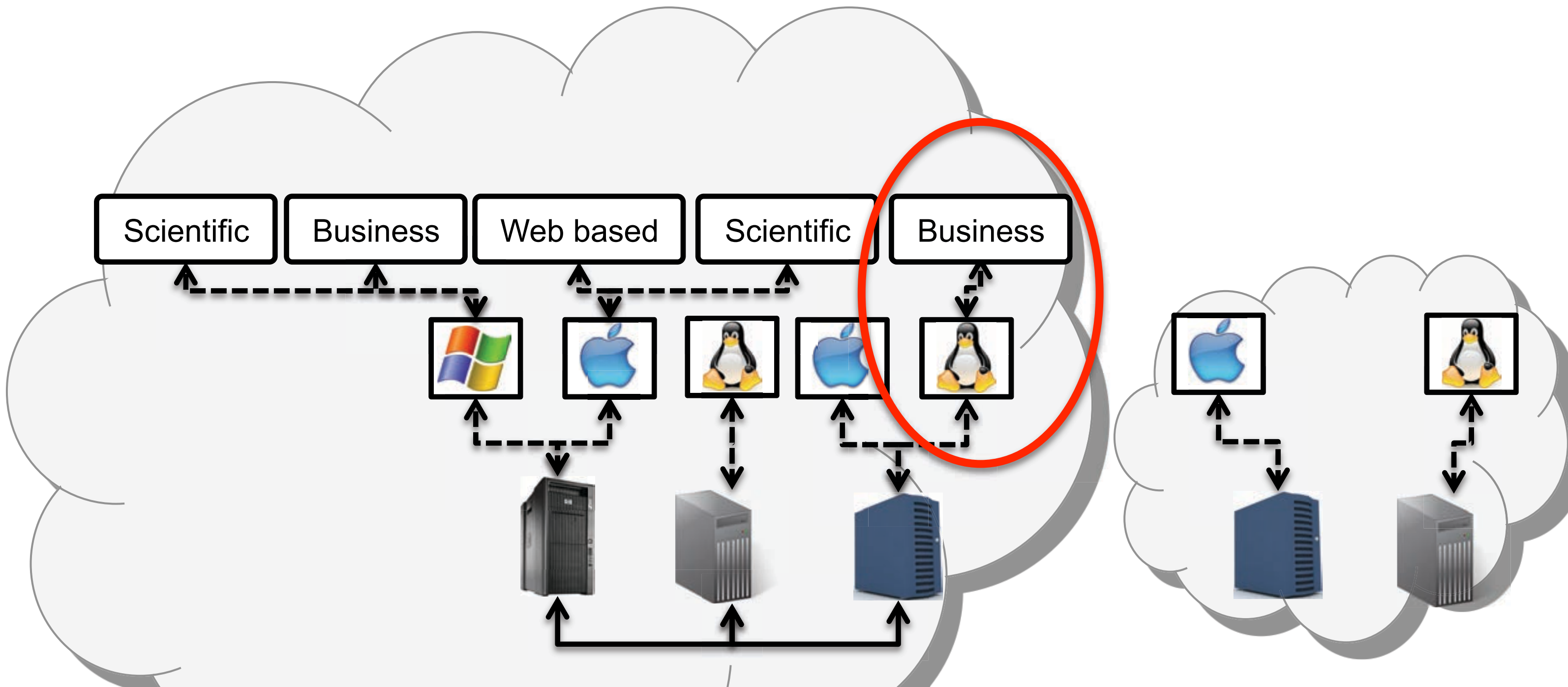
Cloud Computing and Energy Efficiency



Cloud Computing and Energy Efficiency

TU Vienna

Barcelona, Spain

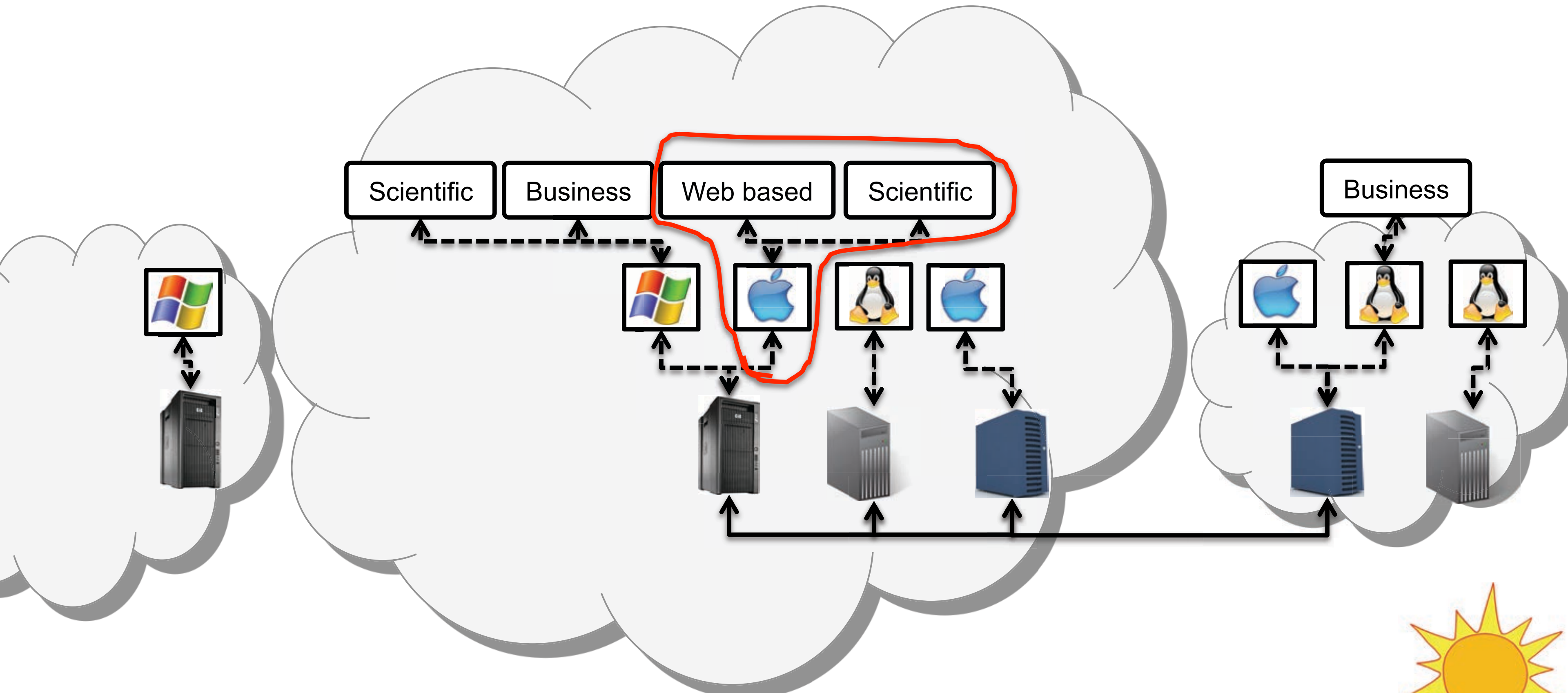


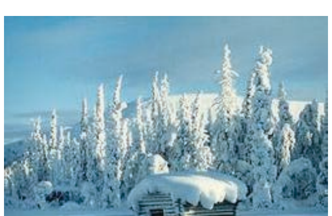
Cloud Computing and Energy Efficiency

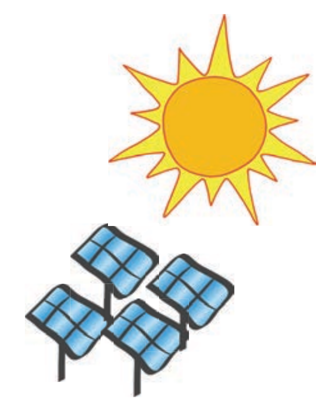
Helsinki, Finland

TU Vienna

Barcelona, Spain

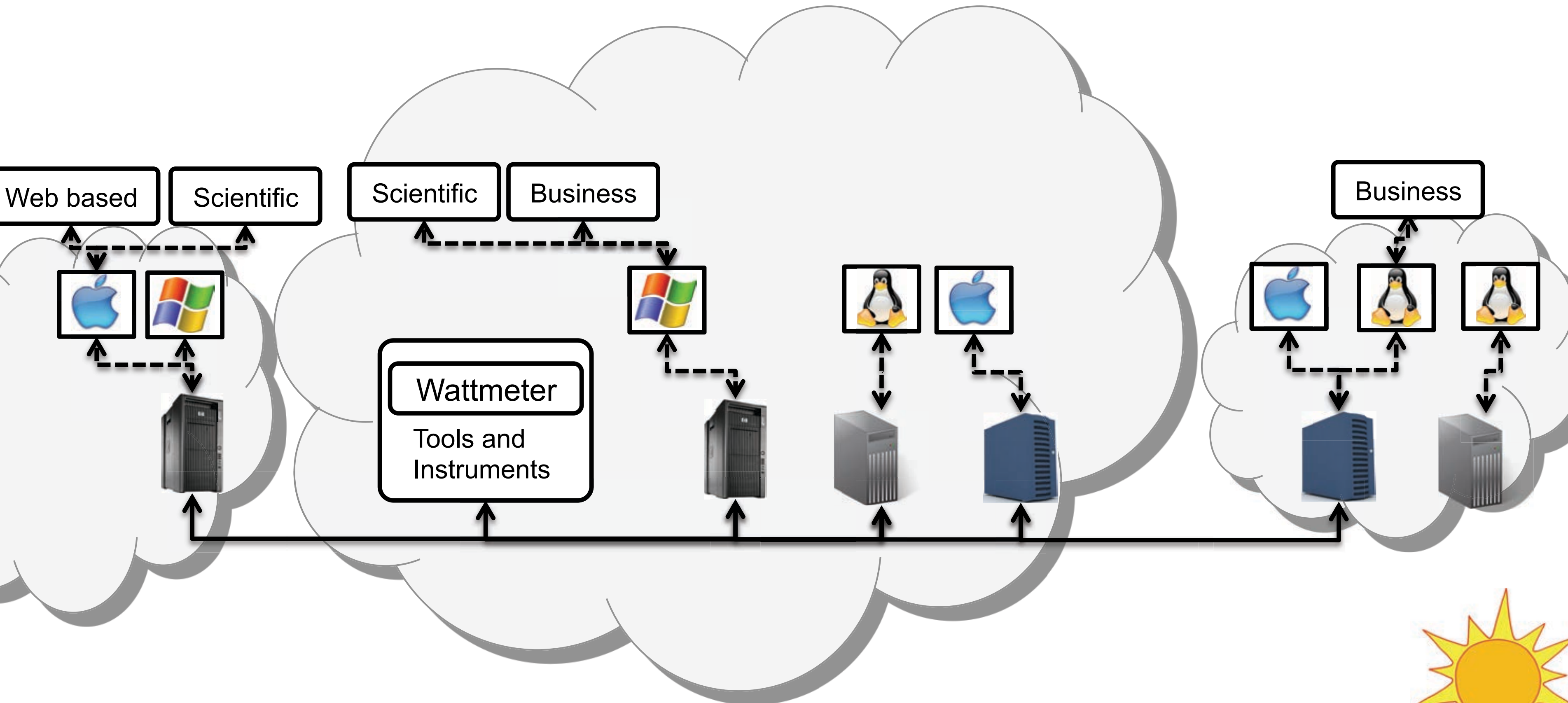


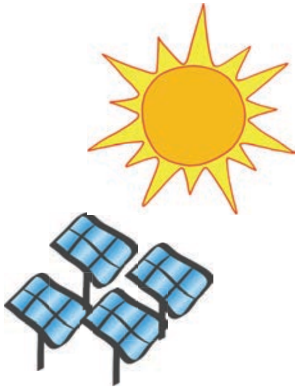




Cloud Computing and Energy Efficiency

Helsinki, Finland | TU Vienna | Barcelona, Spain





Cloud Future: The 5th Utility

product lines management?

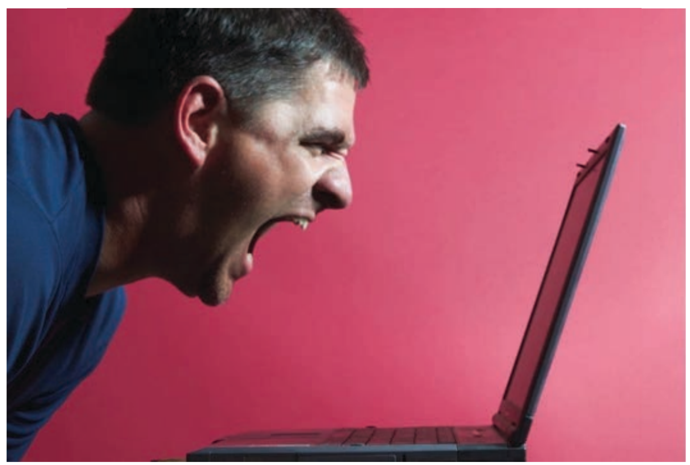
provider lock in?

disaster recovery?

security?

business-models?

privacy?



systematic testing?

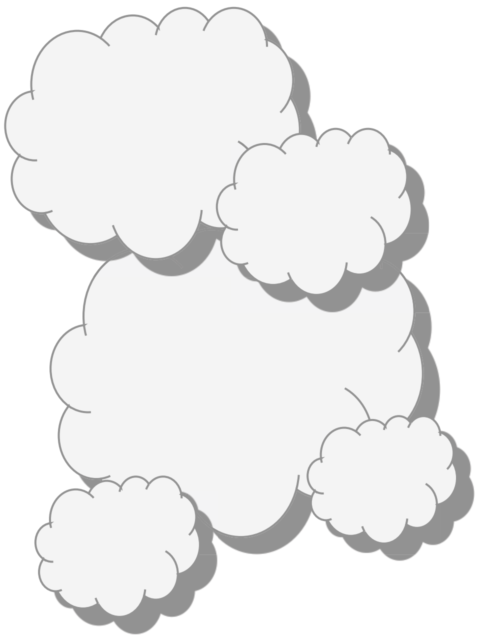
self-configuration?

cost / break even models?

… ?

Source:
Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. ***Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility***, Future Generation Computer Systems, 25(6):599-616, June 2009**. *1434 Citations (Google Scholar)***

Clouds + Big Data



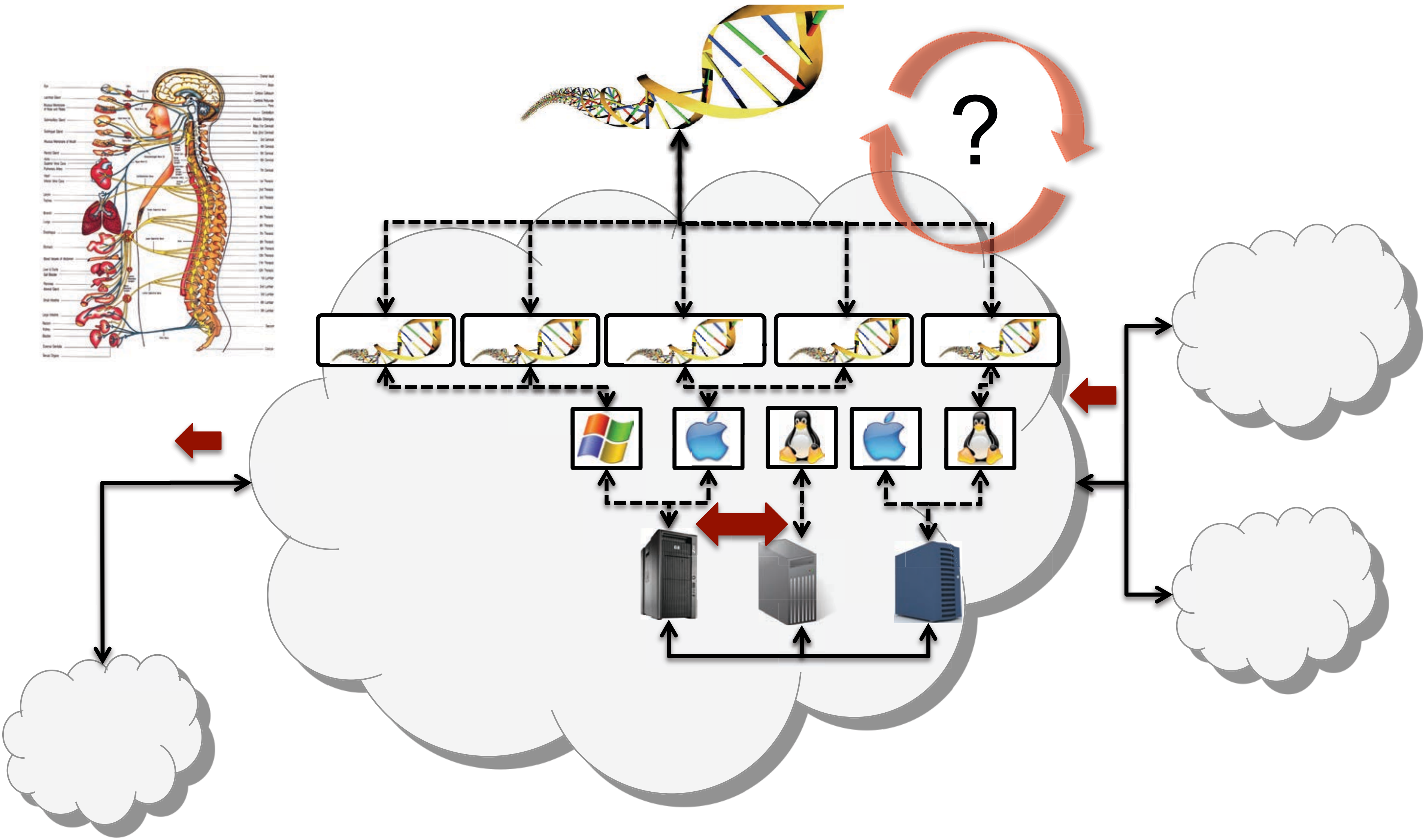
+

=

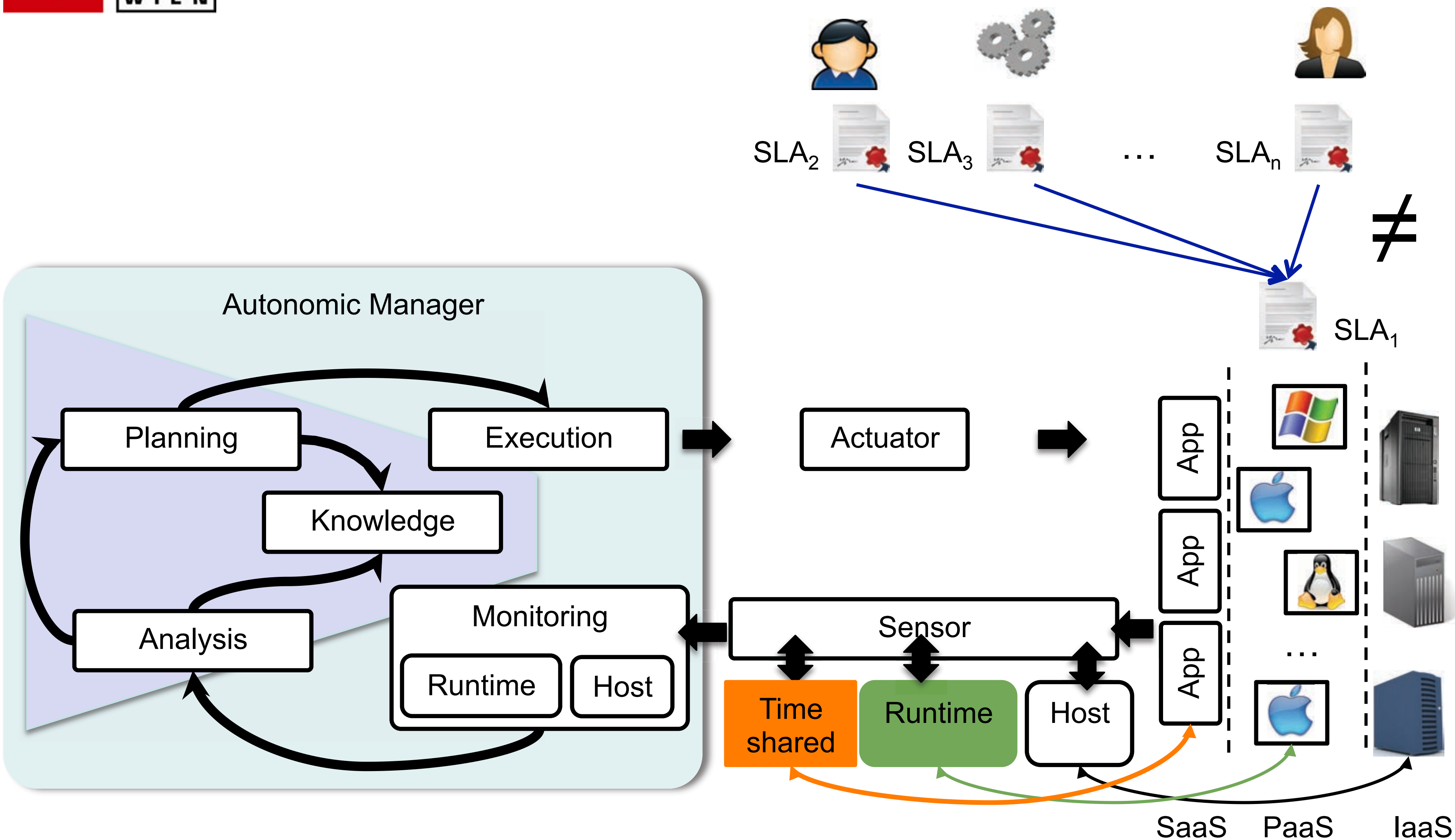
BIG MESS?

ORGANIZE YOUR CLOUDS!

… like some efficient systems in nature …



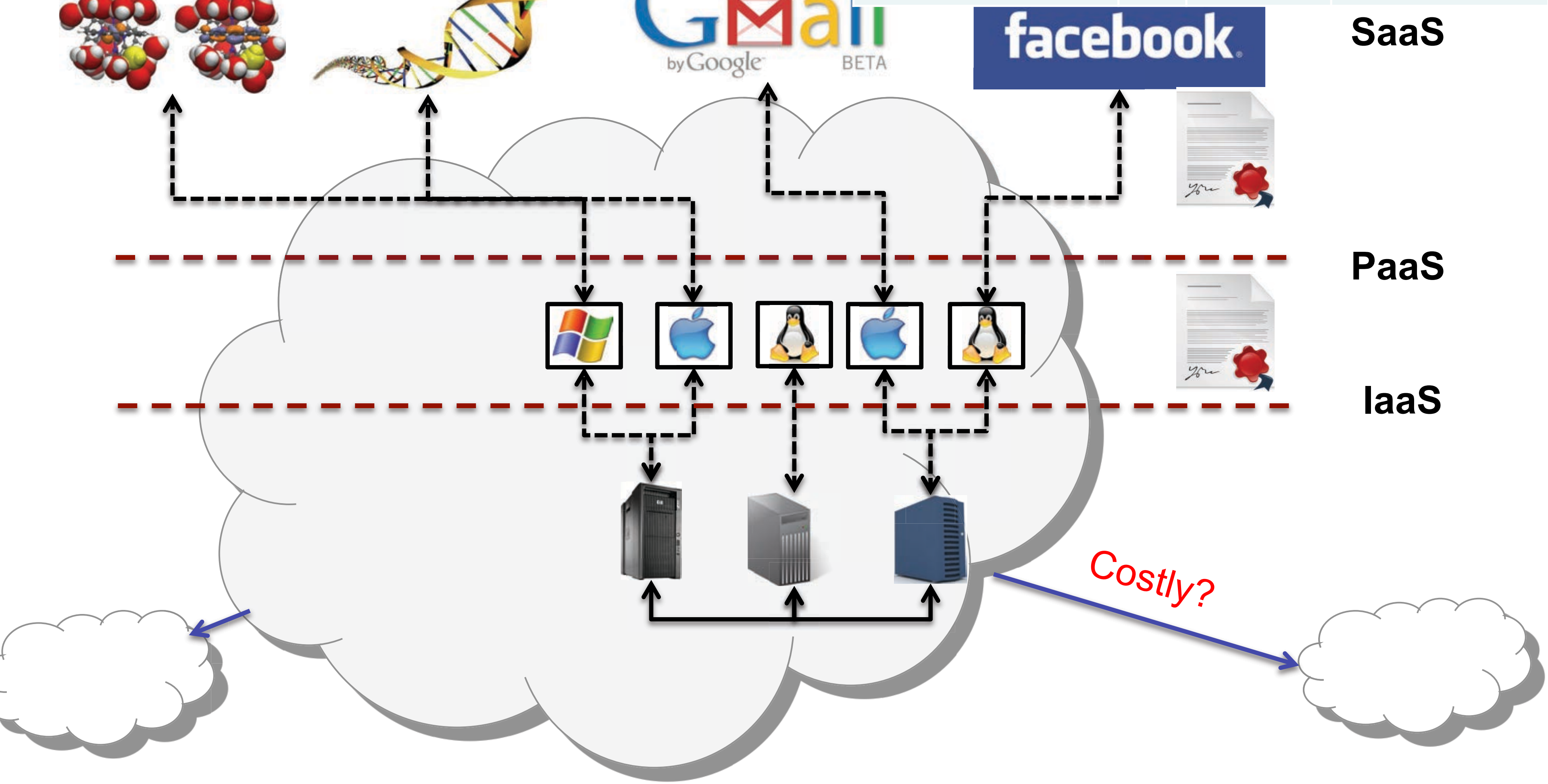
Self-manageable Cloud Services



Project Foundations of self-adaptable ICT Infrastructures

Guaranteeing SLA

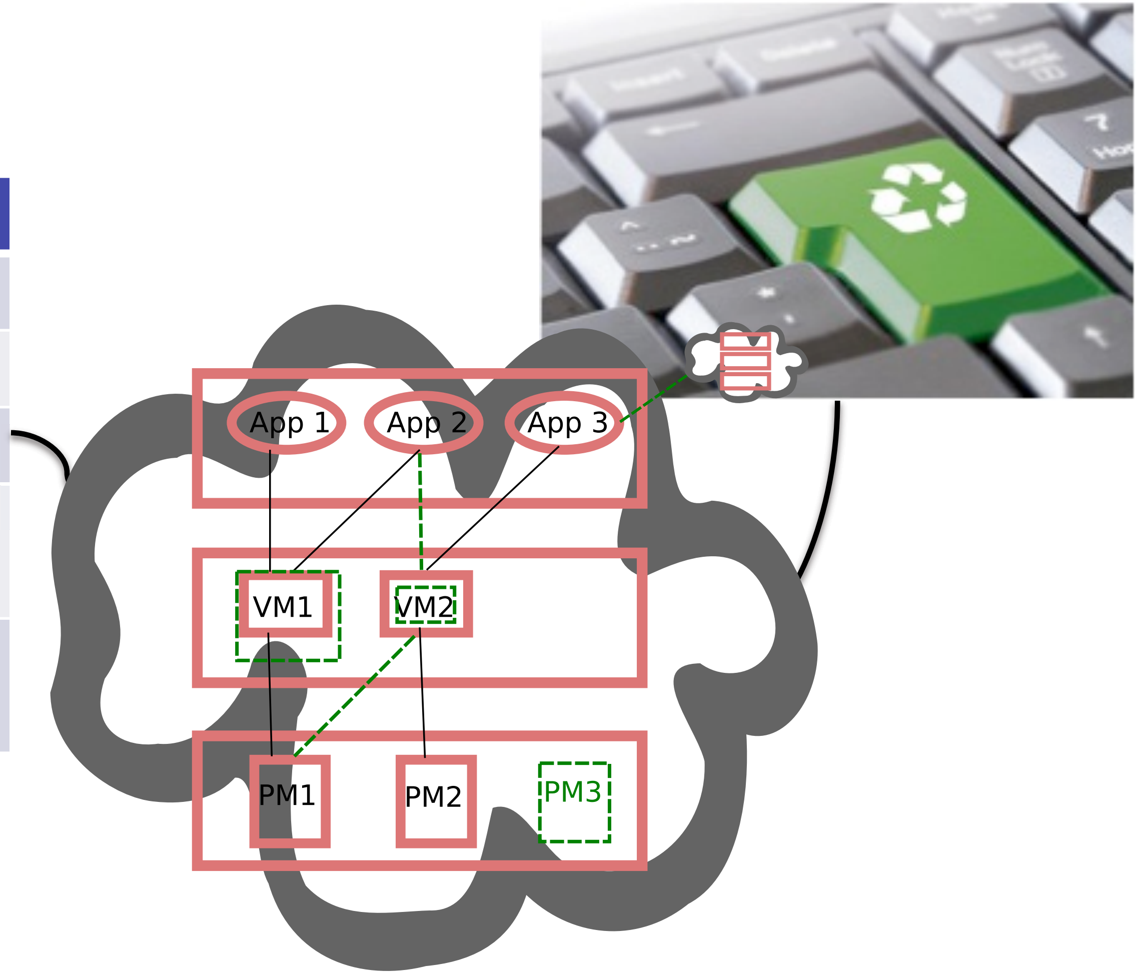
Service Level Agreement (SLA)			
CPU Power	≥	512	MIPS
Memory	≥	1024	MB
Storage	≥	1000	GB
Incoming Bandwidth	≥	10	Mbit/s
Outgoing Bandwidth	≥	20	Mbit/s



Self-adaptability of Cloud Computing Systems?

Challenges

Service Level Agreement (SLA)			
CPU Power	≥	512	MIPS
Memory	≥	1024	MB
Storage	≥	1000	GB
Incoming Bandwidth	≥	10	Mbit/s
Outgoing Bandwidth	≥	20	Mbit/s



Cloud characteristics:

- dynamic
- on demand: computing as utility
- unforeseen load changes
- autonomic adaptation and (re-) provisioning of resources
- very scalable

Two conflicting goals:

1. Minimize SLA violations
2. Maximize energy efficiency

\+

Achieve 1. and 2. by as few time- and energy consuming reallocation actions as possible

Speculate!

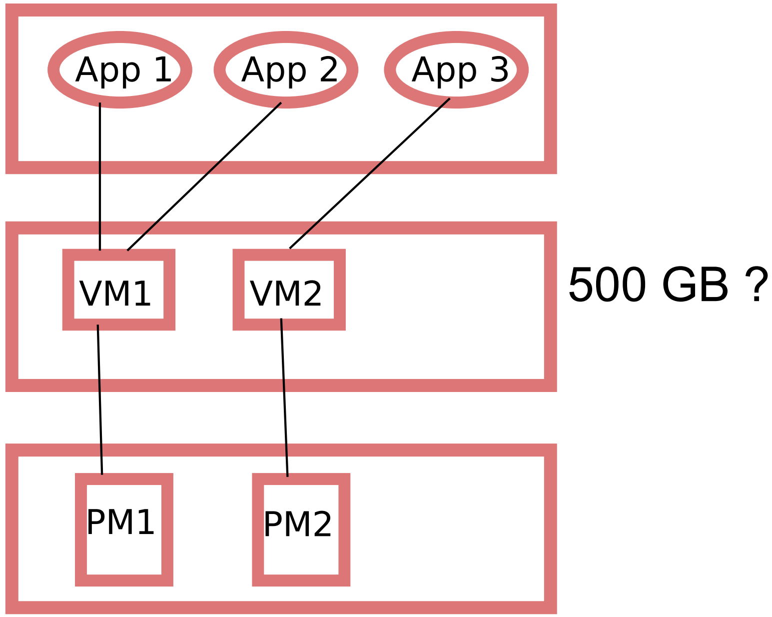
Service Level Agreement (SLA)			
CPU Power	≥	512	MIPS
Memory	≥	1024	MB
Storage	≥	1000	GB
Incoming Bandwidth	≥	10	Mbit/s
Outgoing Bandwidth	≥	20	Mbit/s

Speculative approach:
May we allocate
less resources then agreed,
but more than actually utilized
at the specific point in time –
and not violate SLAs?

Storage			
What do we provide?	What does the consumer utilize?	What was agreed in the SLA?	Violation?
500 GB	400 GB	>= 1000 GB	**NO**
500 GB	510 GB	>= 1000 GB	**YES**
1000 GB	1010 GB	>= 1000 GB	**NO**

How to structure actions?

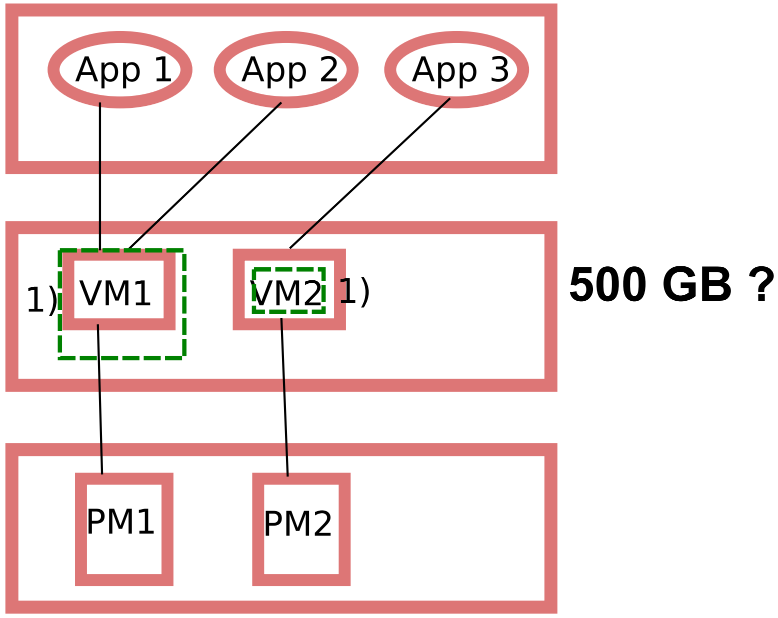
- Escalation levels:



How to structure actions?

- Escalation levels:

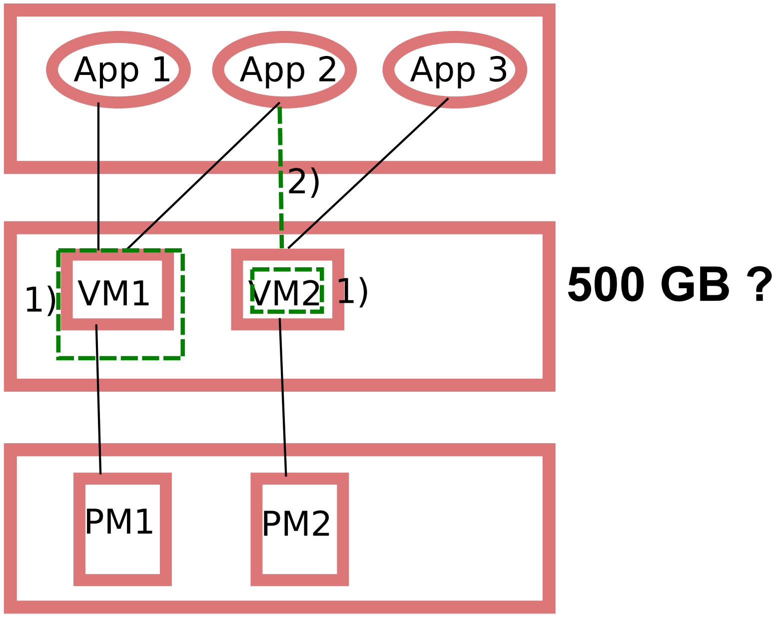
1. Change VM configuration



How to structure actions?

- Escalation levels:

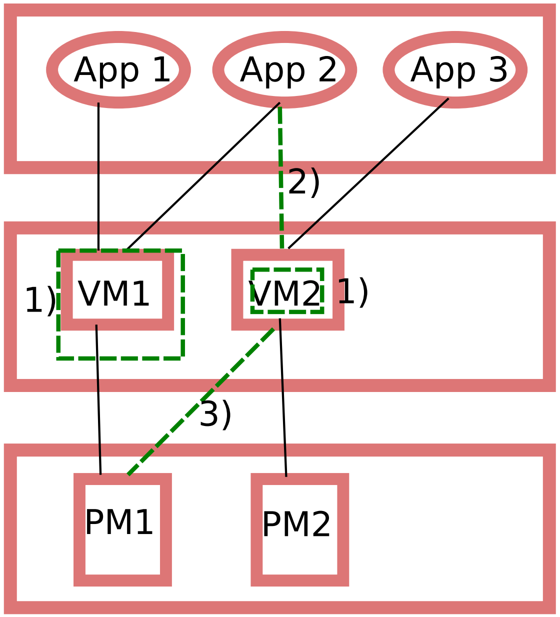
1. Change VM configuration
2. Migrate applications from one VM to another.



How to structure actions?

- Escalation levels:

1. Change VM configuration
2. Migrate applications from one VM to another.
3. Migrate one VM from one PM to another or create new VM on appropriate PM.

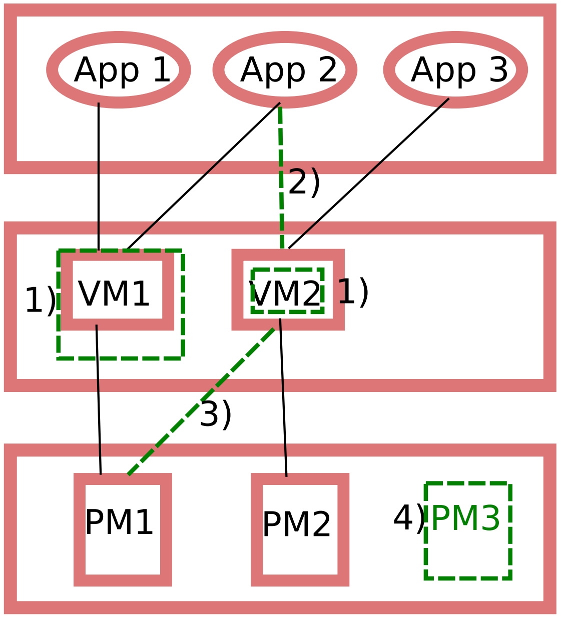


500 GB ?

How to structure actions?

- Escalation levels:

1. Change VM configuration
2. Migrate applications from one VM to another.
3. Migrate one VM from one PM to another or create new VM on appropriate PM.
4. Turn on/off PM.

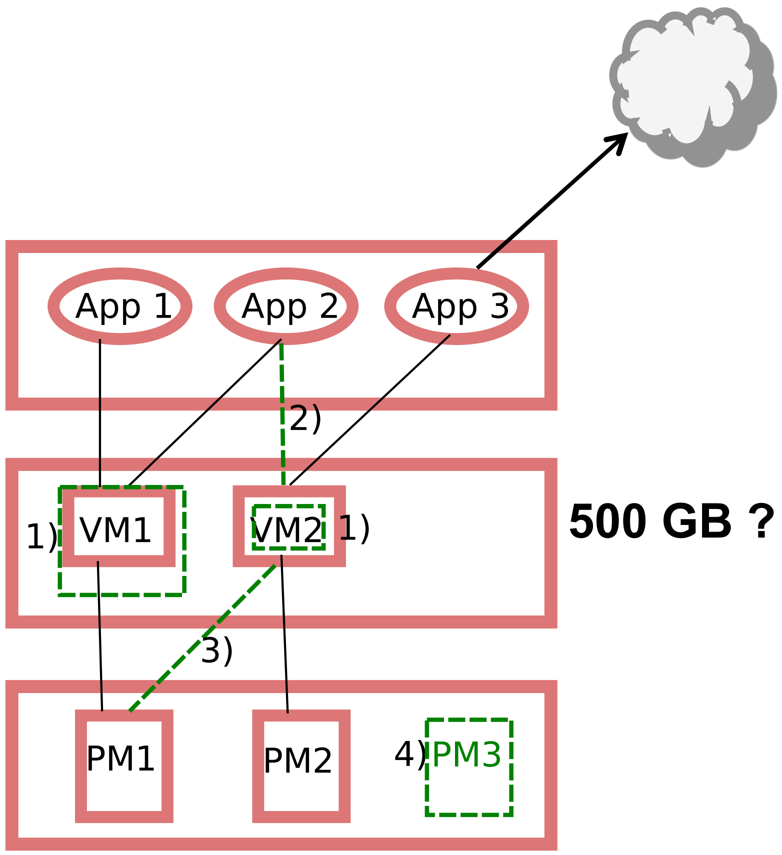


500 GB ?

How to structure actions?

- Escalation levels:

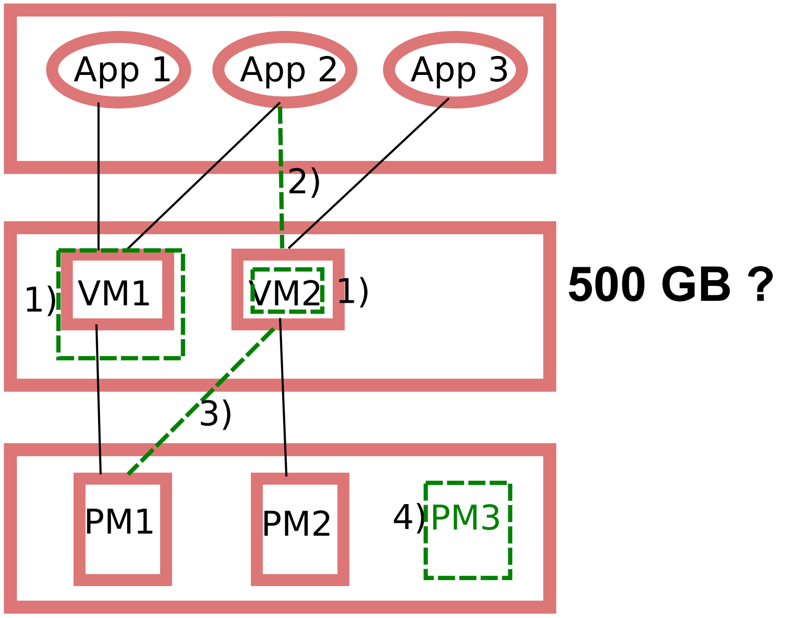
1. Change VM configuration
2. Migrate applications from one VM to another.
3. Migrate one VM from one PM to another or create new VM on appropriate PM.
4. Turn on/off PM.
5. Outsource to other Cloud provider.



How to structure actions?

- Escalation levels:

1. Change VM configuration
2. Migrate applications from one VM to another.
3. Migrate one VM from one PM to another or create new VM on appropriate PM.
4. Turn on/off PM.
5. Outsource to other Cloud provider.
6. **Do nothing!**



How to structure actions?

- Escalation levels:

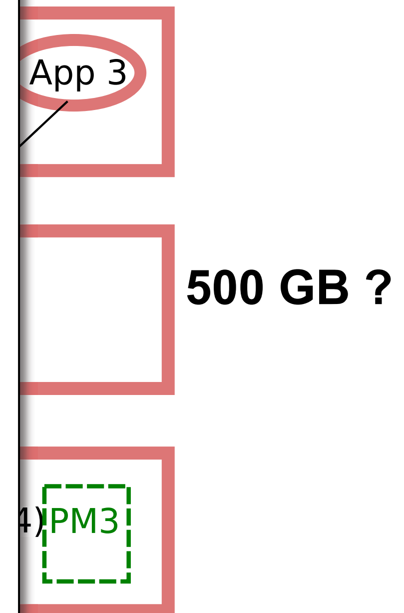
1. Change
2. Migrate
   one VM
3. Migrate
   PM to a
   new VM
   PM.
4. Turn on
5. Outsour
   provider.
6. **Do nothing!**

Methodologies:

- **Rules: IF – THEN**
  - Trend analysis (up, down, oscillations, volatility, …)
- **Case Based Reasoning (CBR)**
- Situation Calculus
- …

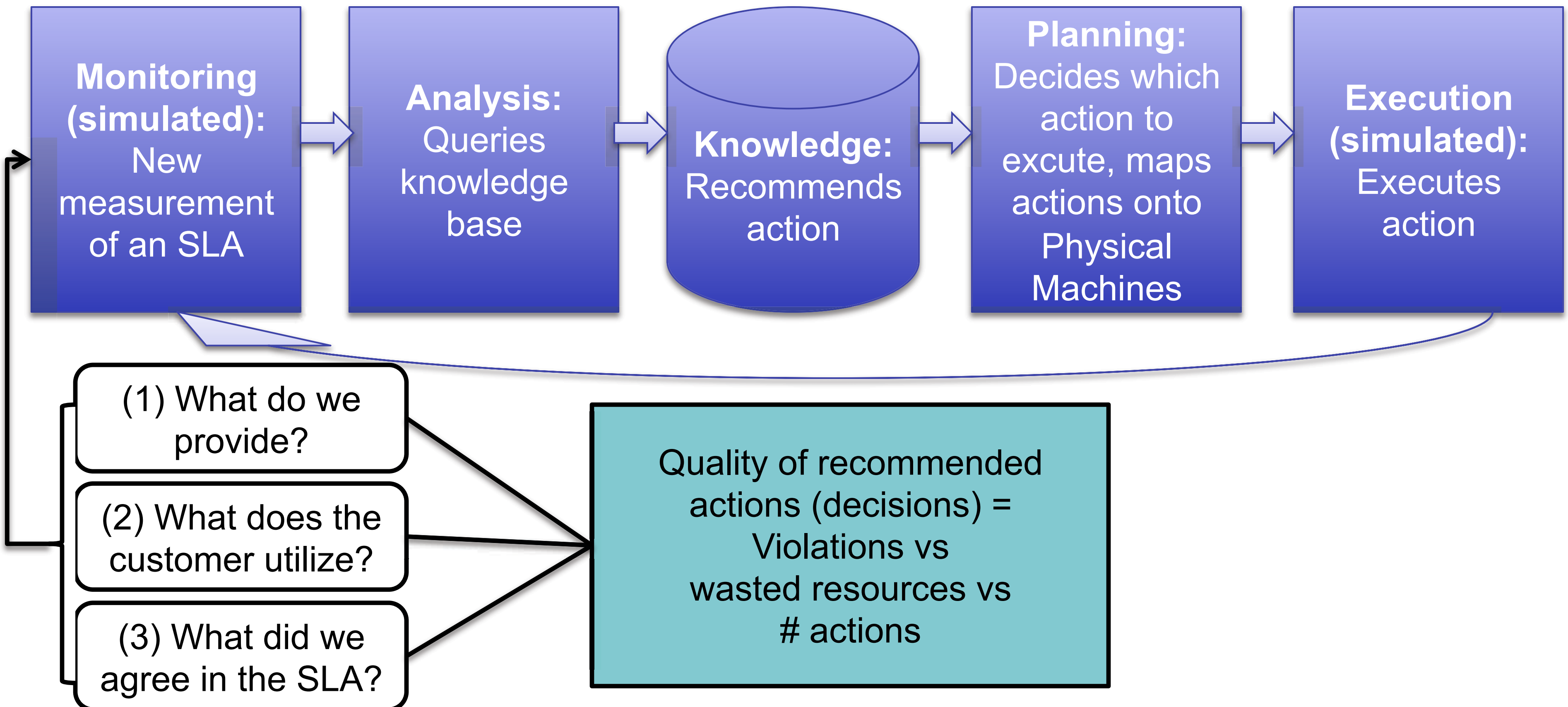
Which one should we use?

Quality of recommended actions (decisions) =
Violations vs.
Wasted resources vs.
Number of actions



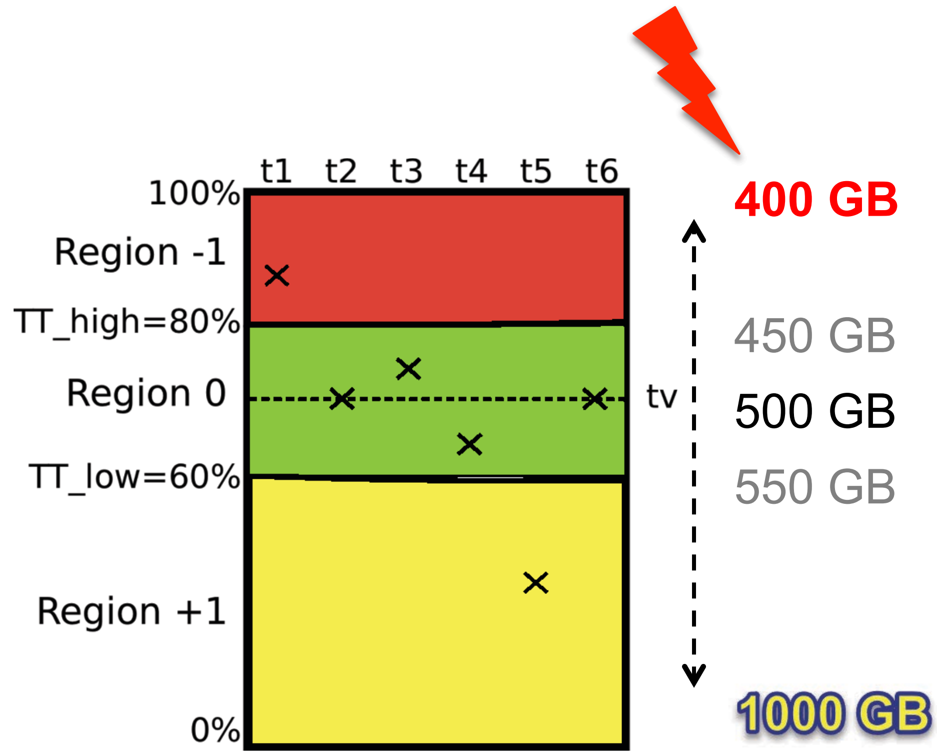
How to evaluate KM techniques?

- Generic simulation engine
  - implements the MAPE-K loop
  - simulates monitoring and execution parts of it.



Energy efficiency vs. SLAs

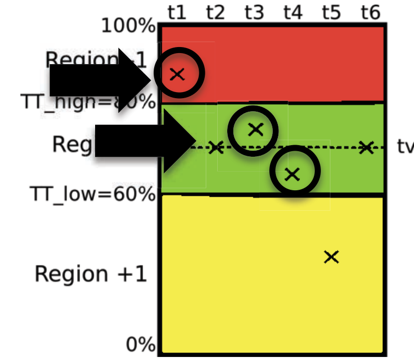
- Knowledge Management
  - Iterative approach
  - keeps representation and characteristics of relevant objects in the Cloud
  - Threat Thresholds (TTs)
  - Over- and under-provisioning



Cloud “entity”

Credits: *M. Maurer*

Rules



- Trend analysis (up, down, oscillations, volatility, …)

- Rules using Drools
  - Rule increasing

Costs

Volatility

1. **IF**
2. $ut^r > TT^r_{\text{high}}$ AND $ut^r_{predicted} > TT^r_{\text{high}}$
3. **THEN**
4. Set pr^r to $\frac{\text{use}^r}{tv(r)}$ if there are plenty of resources
5. Set pr^r to $\min(\frac{\text{use}^r}{tv(r)}, SLO^r * (1 + \epsilon/100))$ if resources are becoming scarce

  - Rule decreasing

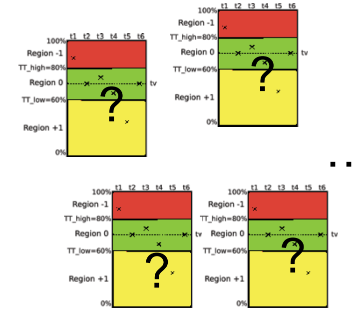
1 **IF**

2 $ut^r < TT^r_{\text{low}}$ AND $ut^r_{predicted} < TT^r_{\text{low}}$

3 **THEN**

4 Set pr^r to $\max(\frac{\text{use}^r}{tv(r)}, minPr^r)$.

Cost function



- Define **cost function**

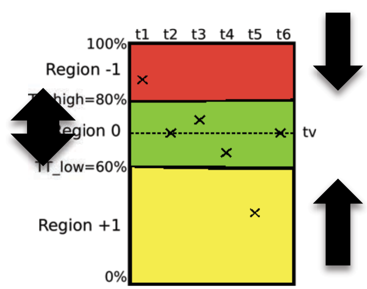
$$c(p,w,c)=\sum_r P^r(p^r)+W^r(w^r)+A^r(a^r)$$

- Cost function - Adapt TTs, if cost increased for last ***k*** iterations
  - Resource ***r*** that had the most cost increasement over the last ***k*** iterations (look-back horizon)
  - Determine most appropriate TT(s) to adapt ? → different options
  - How much should TT be adapted?

$$0\% < TT_{low} < TT_{high} < 100\%$$

$$TT_{low}^{r,t+1}=TT_{low}^{r,t}-\frac{TT_{low}^{r,t}}{\alpha}$$

$$TT_{high}^{r,t+1}=TT_{high}^{r,t}-\frac{100-TT_{high}^{r,t}}{\alpha}$$

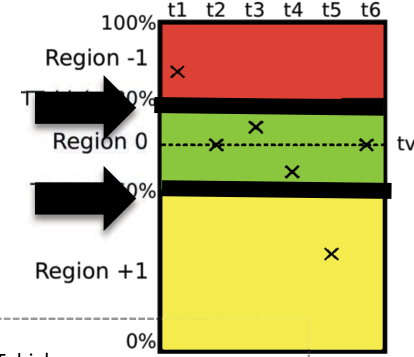


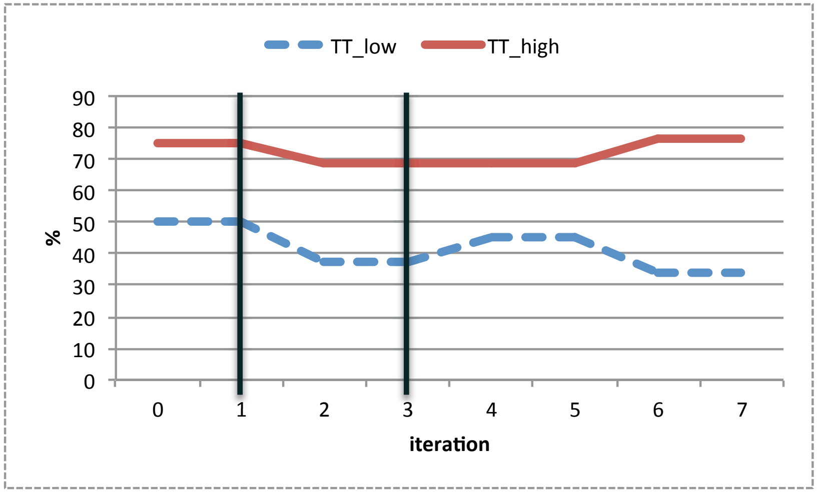
$$TT_{low}^{r,t+1}=TT_{low}^{r,t}-\frac{TT_{high}^{r,t}-TT_{low}^{r,t}}{\alpha}$$

$$TT_{low}^{r,t+1}=TT_{high}^{r,t}-\frac{TT_{high}^{r,t}-TT_{low}^{r,t}}{\alpha}$$

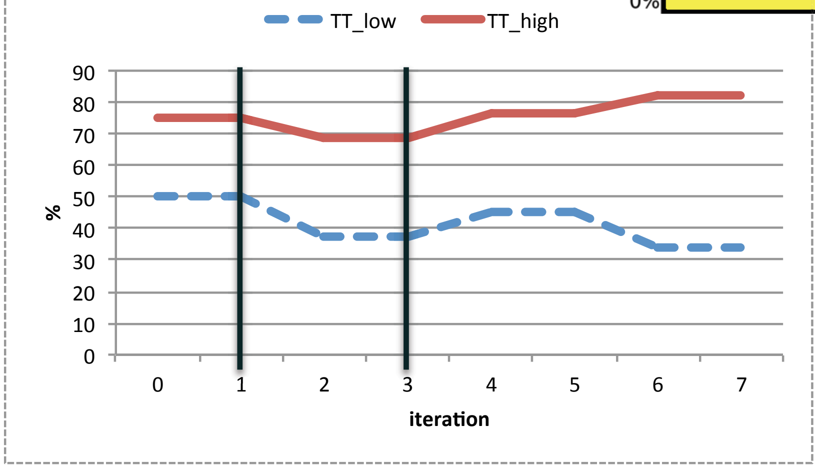
where $1/\alpha<1$ distance from TT_{low} to *0* and TT_{high} to *100*

Dynamic Adaptation of Threat Thresholds





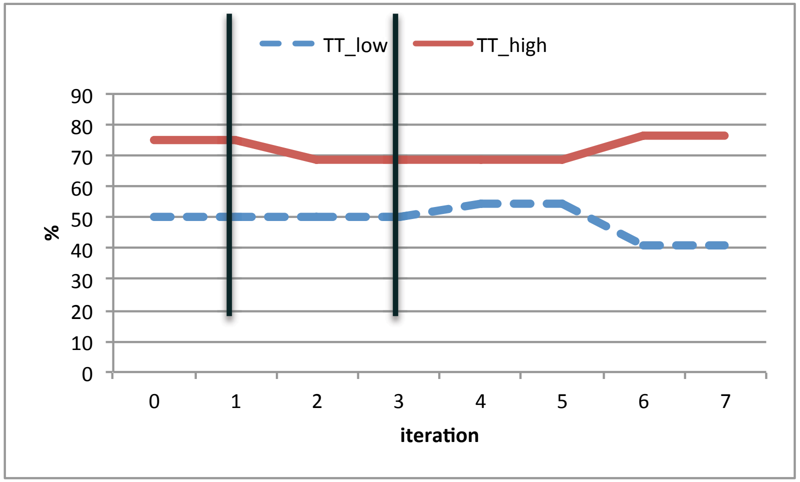
*Minimize SLA Violation (**Option A**)*



*Emphasize Energy Efficiency (**Option B**)*

Maximum in

1) Penalties
2) Wastage
3) Actions



*Very speculative (**Option C**)*

Workload Volatility

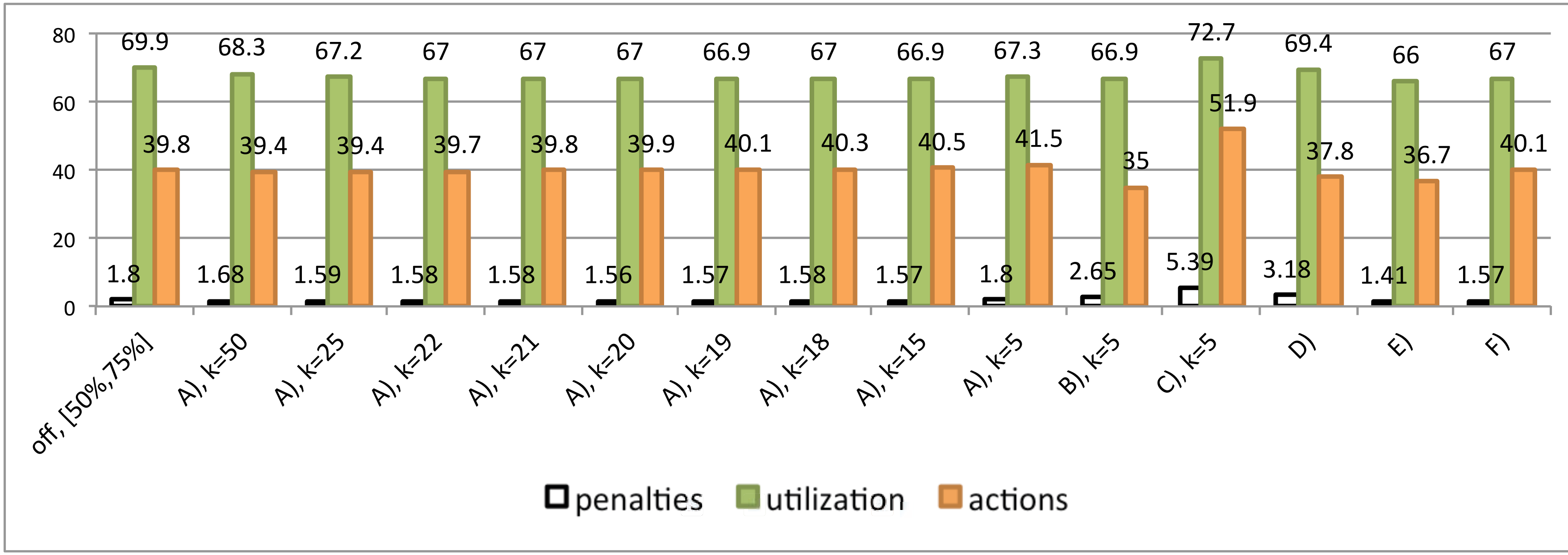
- Define **workload volatility** (WV)

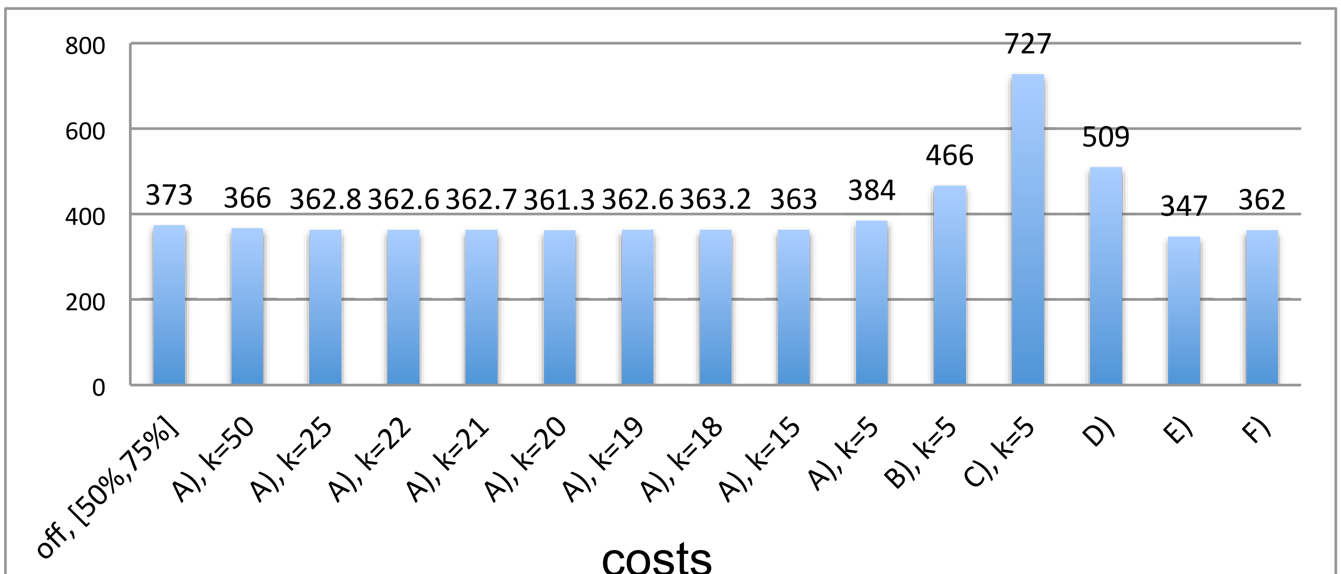
$$\Phi^{r,t}(m^{r,t}, m^{r,t-1}) = \left| \left(\frac{\max(m^{r,t}, r_{\min})}{\max(m^{r,t-1}, r_{\min})} - 1 \right) * 100 \right|$$

- Define WV classes LOW, …, MEDIUM, …, HIGH
- Determine most appropriate TT(s) to adapt ? → different options

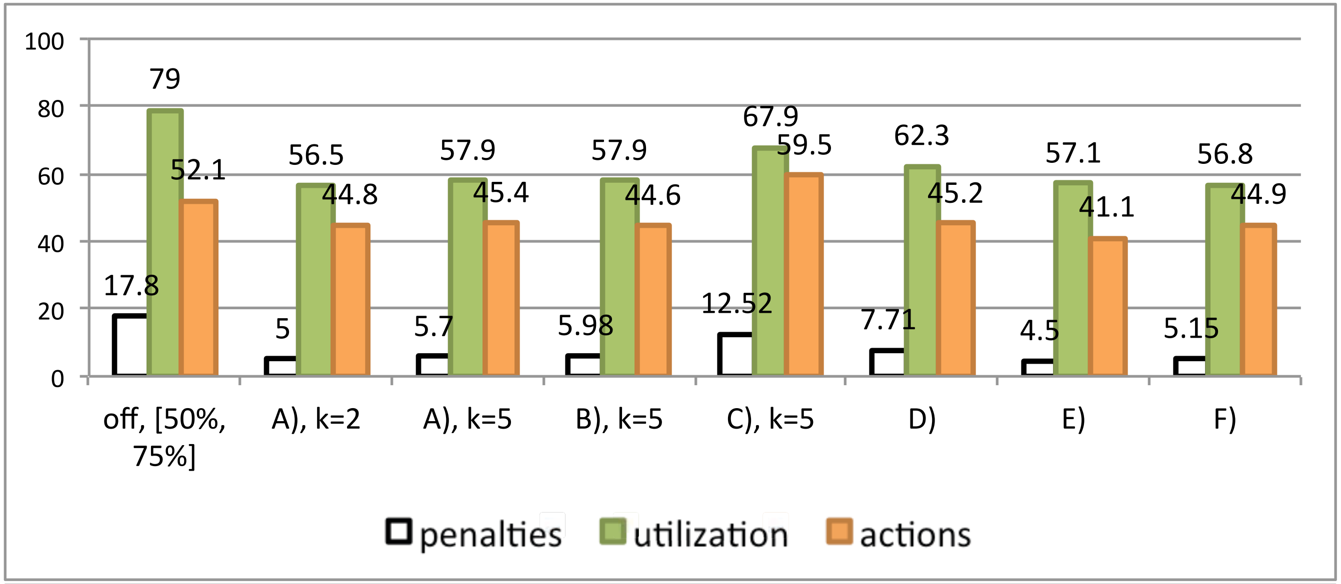
	Option E		*Option F*	
Volatility (Φ)	**TT_{low}**	**TT_{high}**	**Choose Options**	
LOW	70%	90%	C (k=5)	← *Very speculative*
MEDIUM	45%	70%	A (k=20)	← *Minimize SLA Violations*
MEDIUM HIGH	30%	60%	A (k=5)	← *Minimize SLA Violations even more*
HIGH	20%	50%	A (k=2)	← *Be sure to really minimize SLA violations*

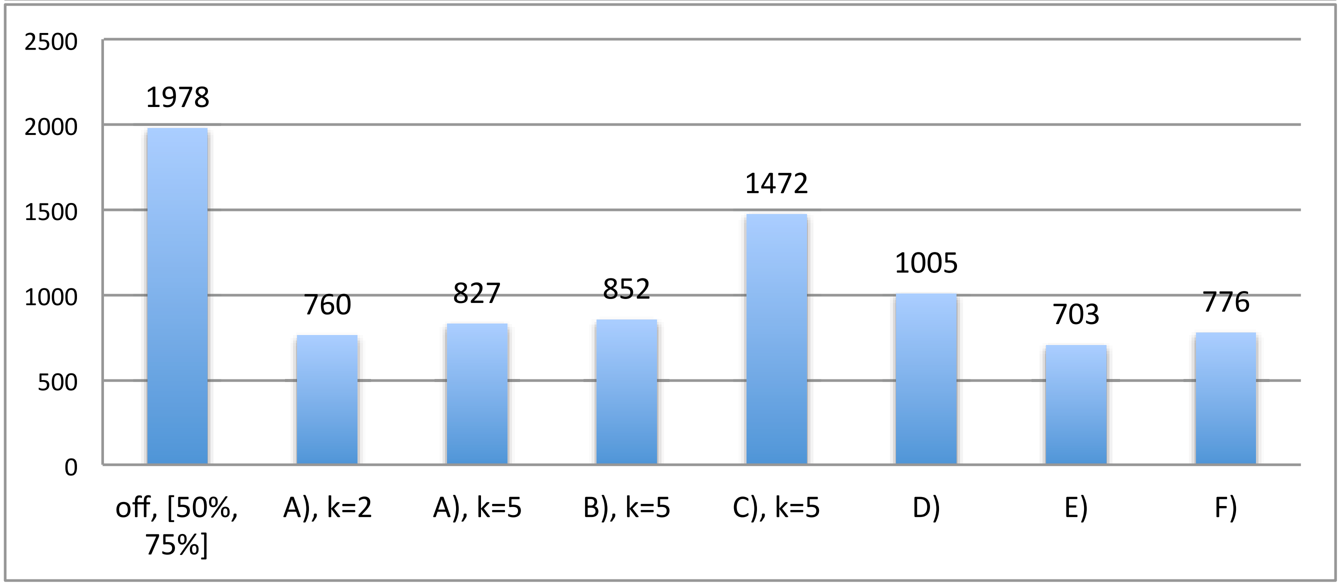
MEDIUM workload volatility





HIGH workload volatility

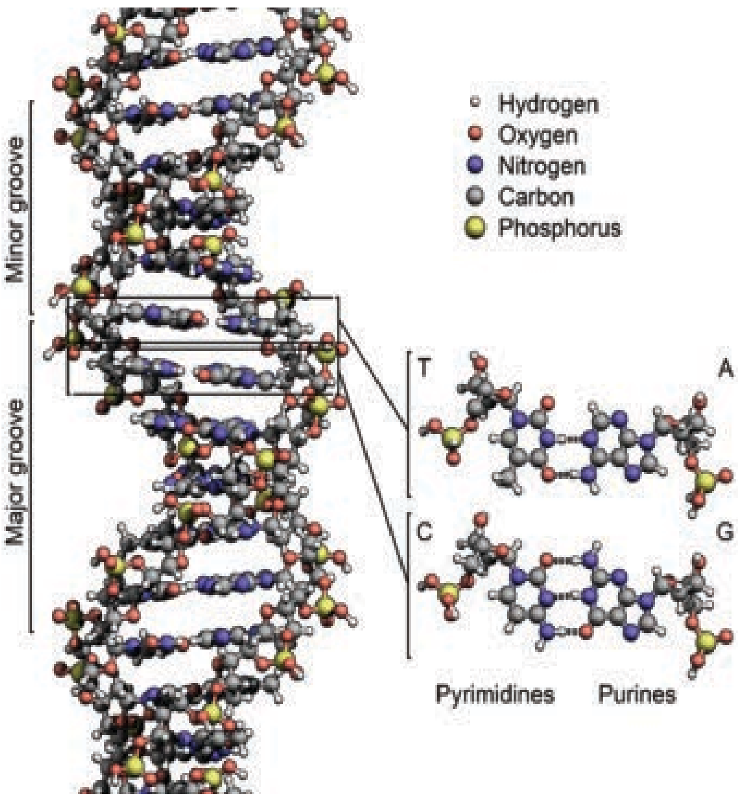




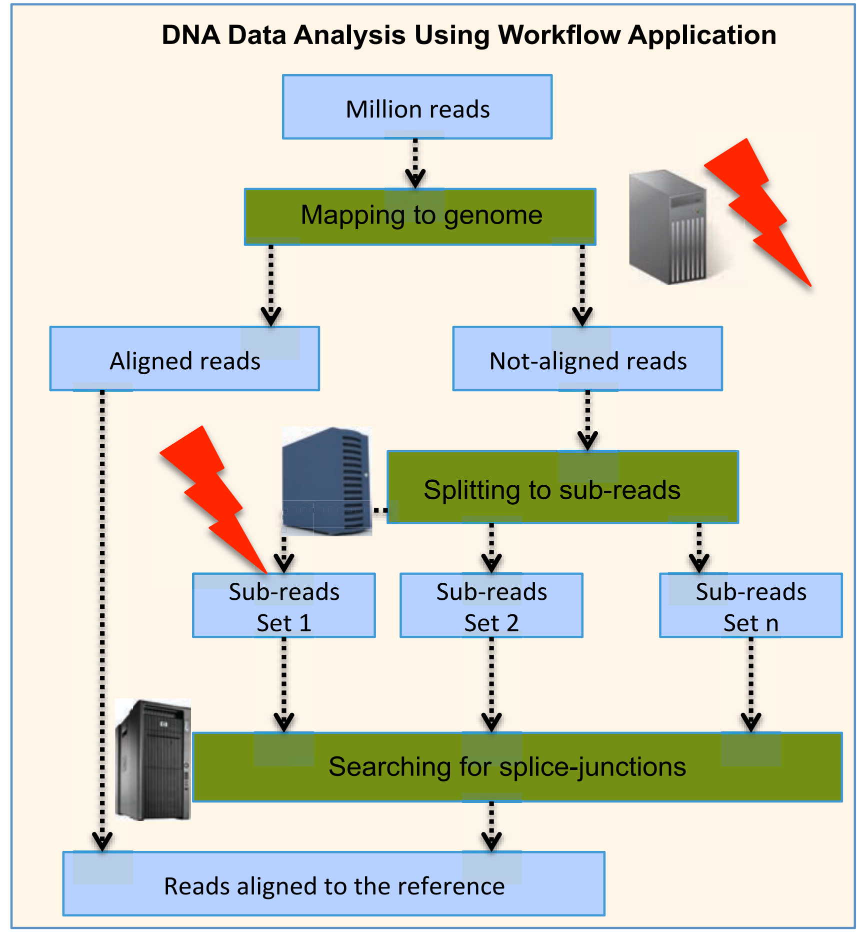
PUT YOUR (BIG) DATA ON THE CLOUD

Use Case: DNA Sequencing Workflows

- Data analysis in bioinformatics



DNA Sequence Sample



Biological Meaning!

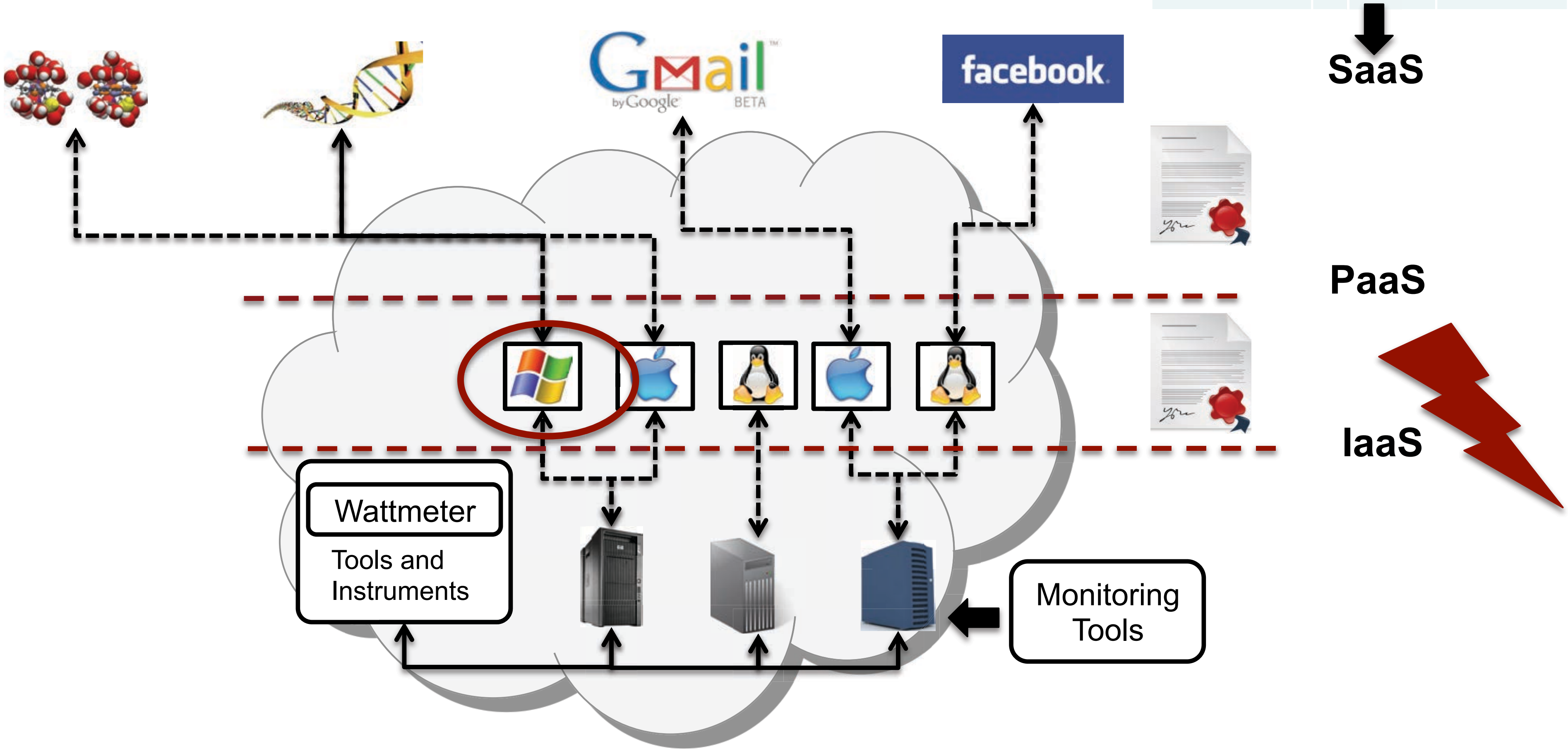
Applications for Next Generation Sequencing

Collaboration with BOKU Vienna (Group D. Kreil)

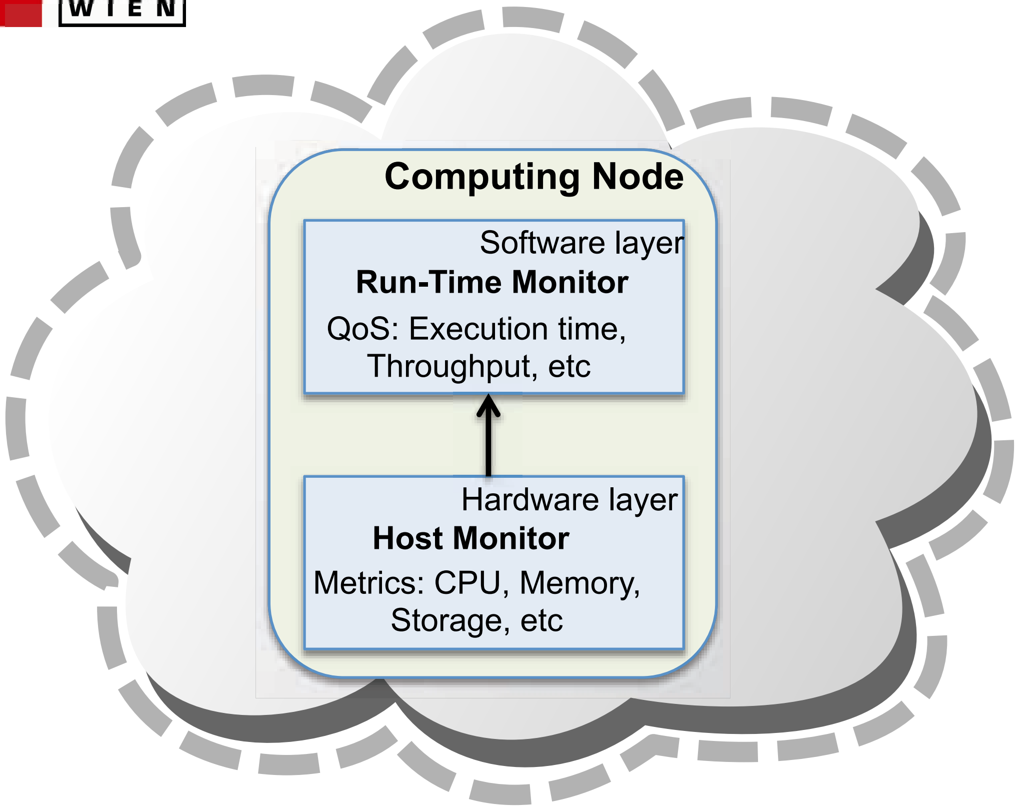
Monitoring

SLA aware and Resource Shared Monitoring

Service Level Agreement (SLA)			
CPU Power	≥	512	MIPS
Memory	≥	1024	MB
Storage	≥	1000	GB
Incoming Bandwidth	≥	10	Mbit/s
Outgoing Bandwidth	≥	20	Mbit/s



Monitoring Technique



$$Availability = \left(1 - \frac{downtime}{uptime}\right) * 100(\%)$$

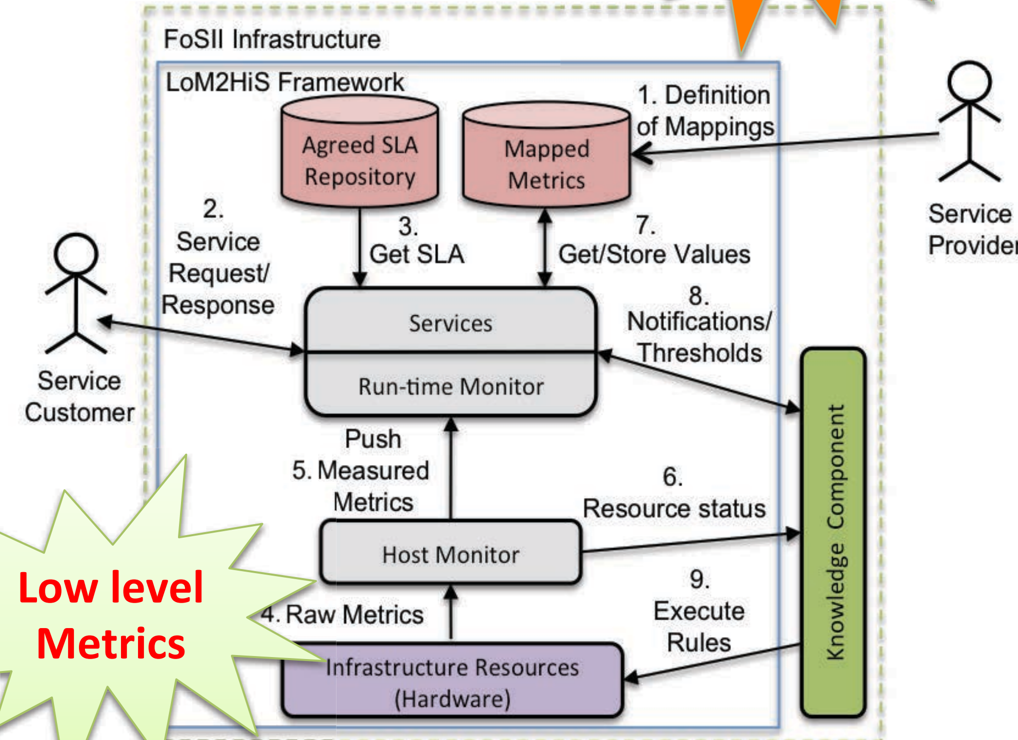
- Monitors **low level resource metrics** (e.g., CPU, uptime, downtime, etc)
- Maps metrics to **Quality of Service parameters** (e.g., response time, throughput, etc)
- Monitoring agents

Credits: V. Emeakaroha

Mapping / Monitoring Framework

High level SLA

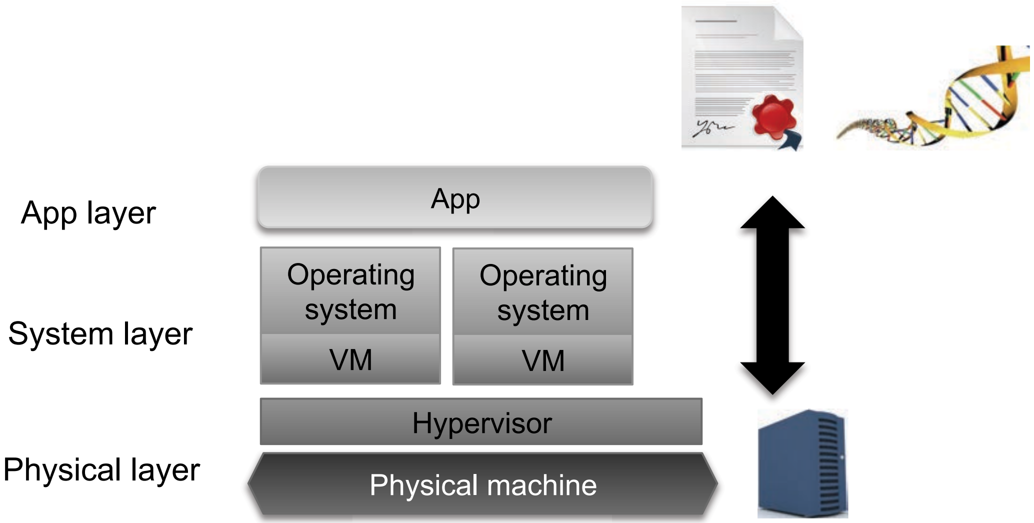
- Low level Metrics to High level SLA
- Comprises of three parts:
  - **Host monitor**
    - Low level metrics
  - **Communication Mechanism**
    - Message transmission
  - **Run-time monitor**
    - Metrics mapping
    - High level service SLA monitoring

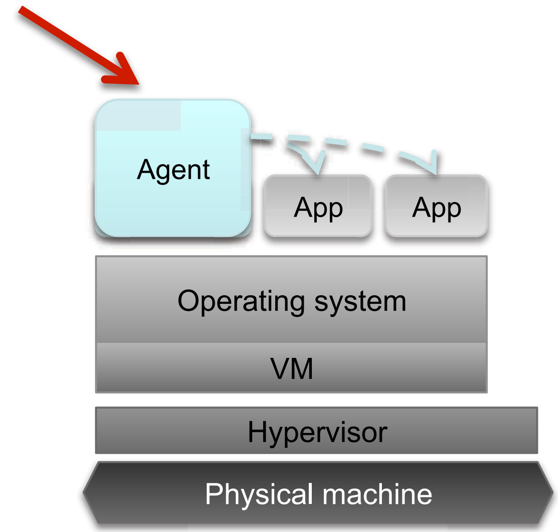


Low level Metrics

Resource Metric	SLA Parameter	Mapping Rule
Uptime, Downtime	Availability (A)	$A = 1 - \frac{downtime}{uptime}$
Inbytes, outbytes, packetsize, avail. Bandwithin, avail. Bandwithout	Response Time (R)	$R = Rin + Rout(ms)$ $Rin = \frac{packetsize}{avail.BandwithIn - Inbytes}$ $Rout = \frac{packetsize}{avail.BandwithOut - Outbytes}$

Lessons Learned: Time sharing of Resources





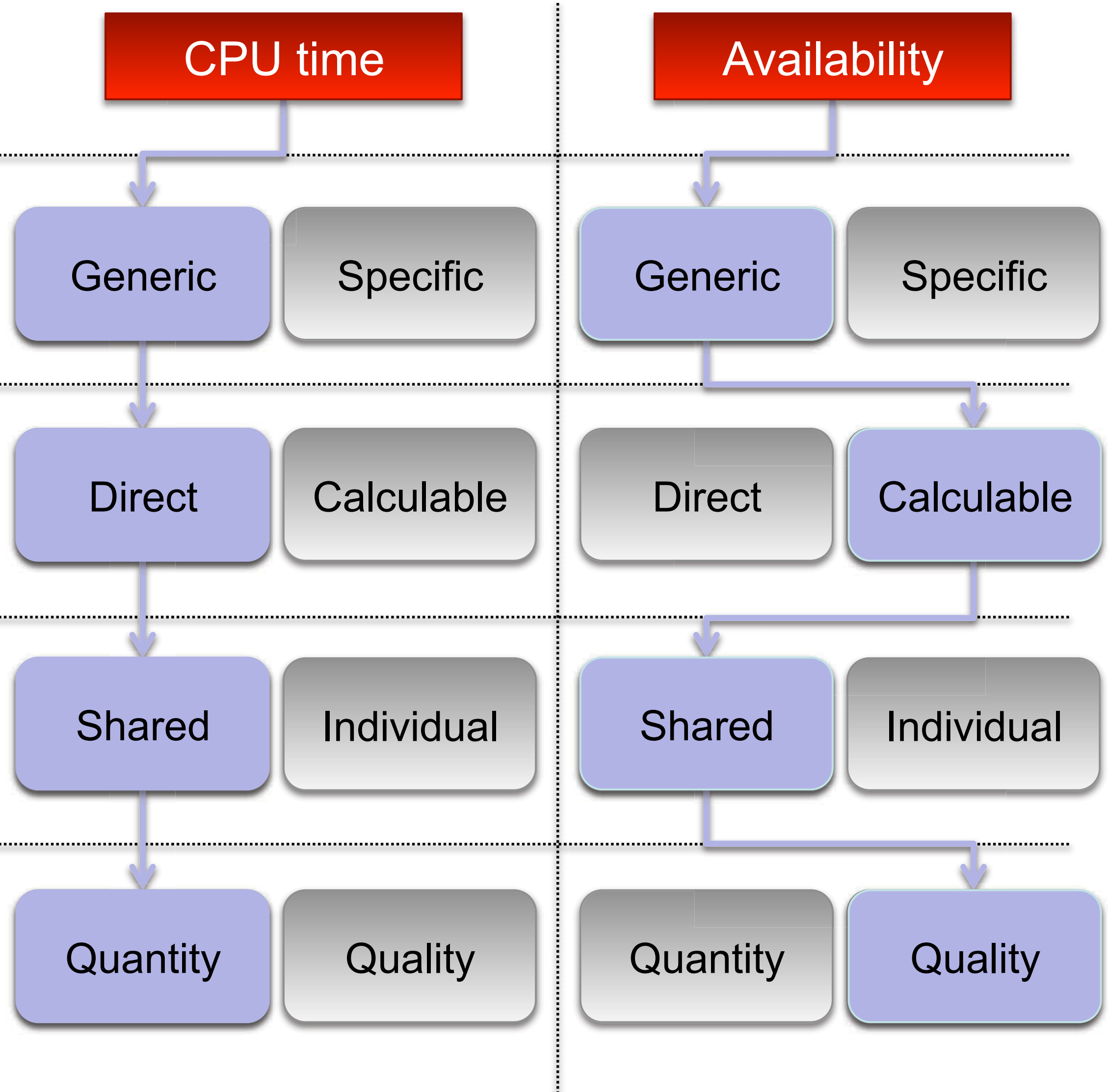
- Monitors **low level resource metrics** (e.g., CPU, uptime, downtime, etc)
- Maps metrics to **Quality of Service parameters** (e.g., response time, throughput, etc)
- Monitoring agents
- Model:
  - **1 PM : n VM** **1 VM -> 1 PM**
  - **1 VM : m Apps** **1 App -> 1 VM**

- Model:
  - **1 PM : n VM** **1 VM -> 1 PM**
  - **1 VM : m Apps** **X App -> 1 VM**

Cloud Metric Classification (CMC)

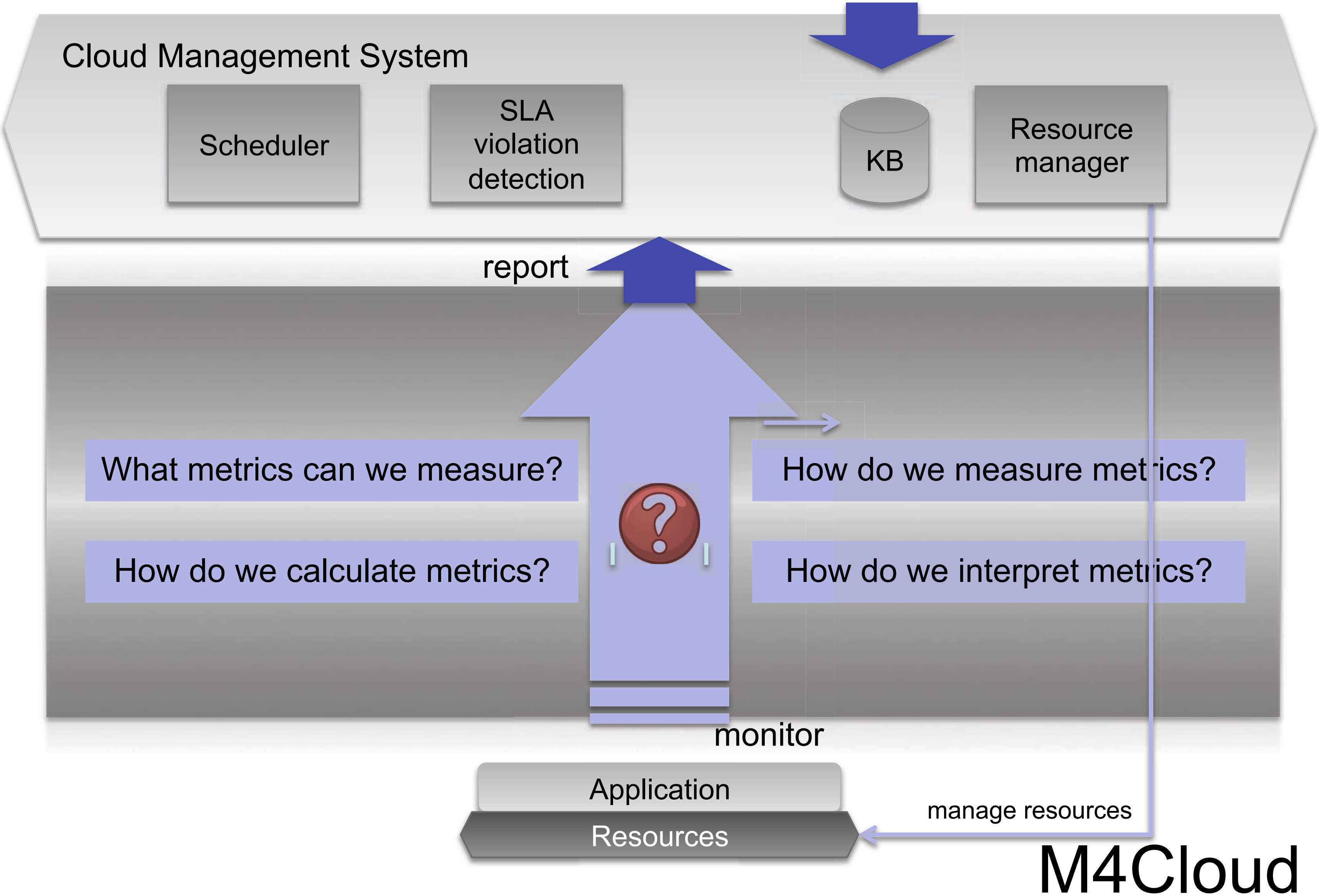
Four (4) models:

- **Application based**
- **Measurement based**
- **Implementation based**
- **Nature based**

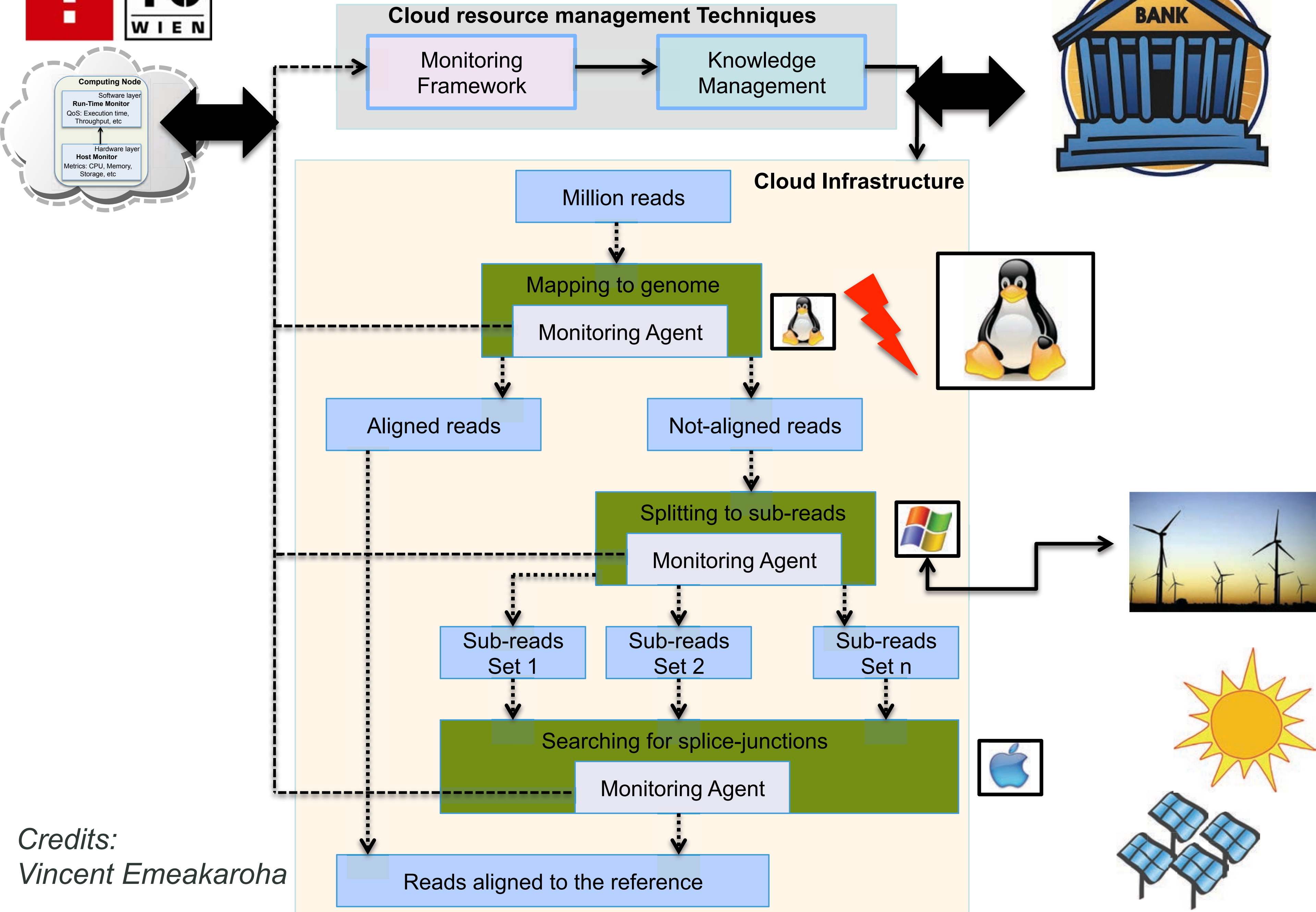


Credits: T. Mastelic

M4Cloud model



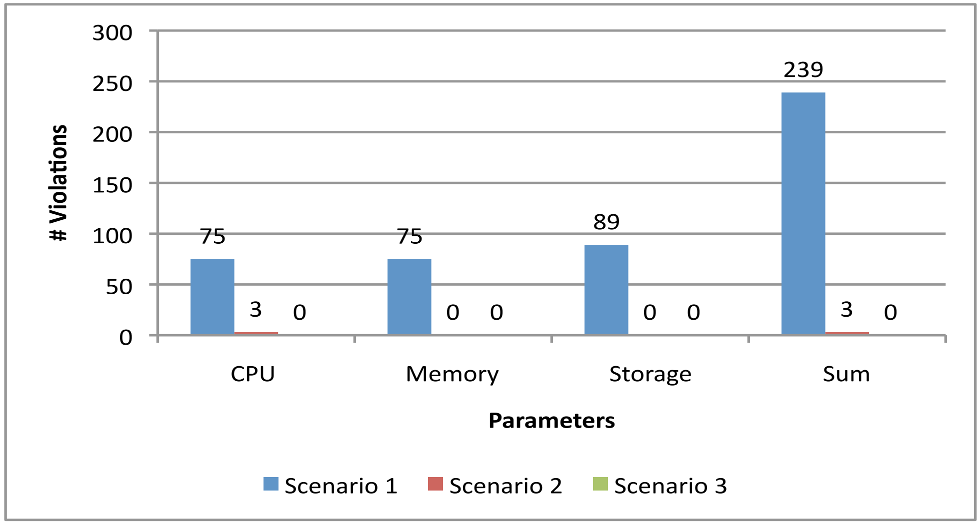
Optimizing Workflow Executions

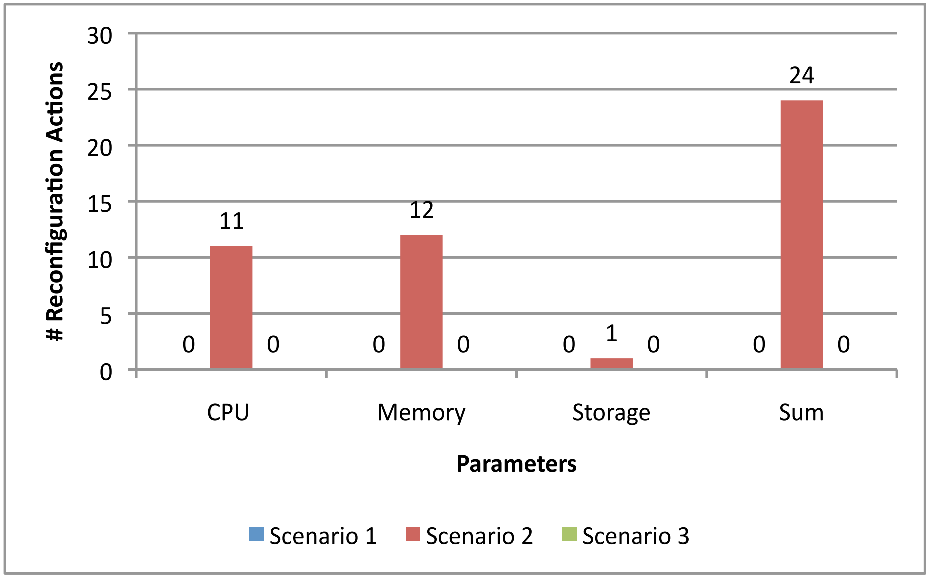


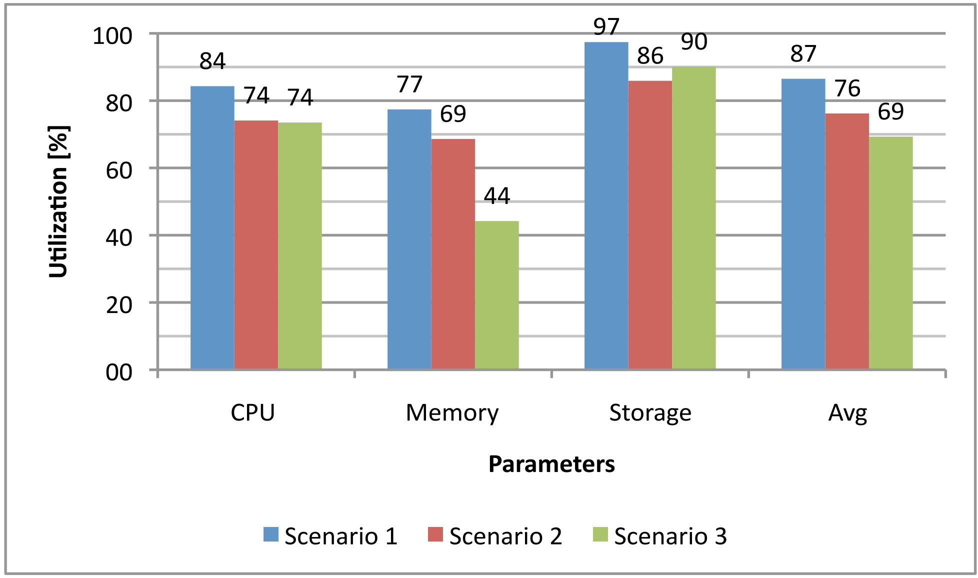
Credits:
Vincent Emeakaroha

Resource Allocation Results

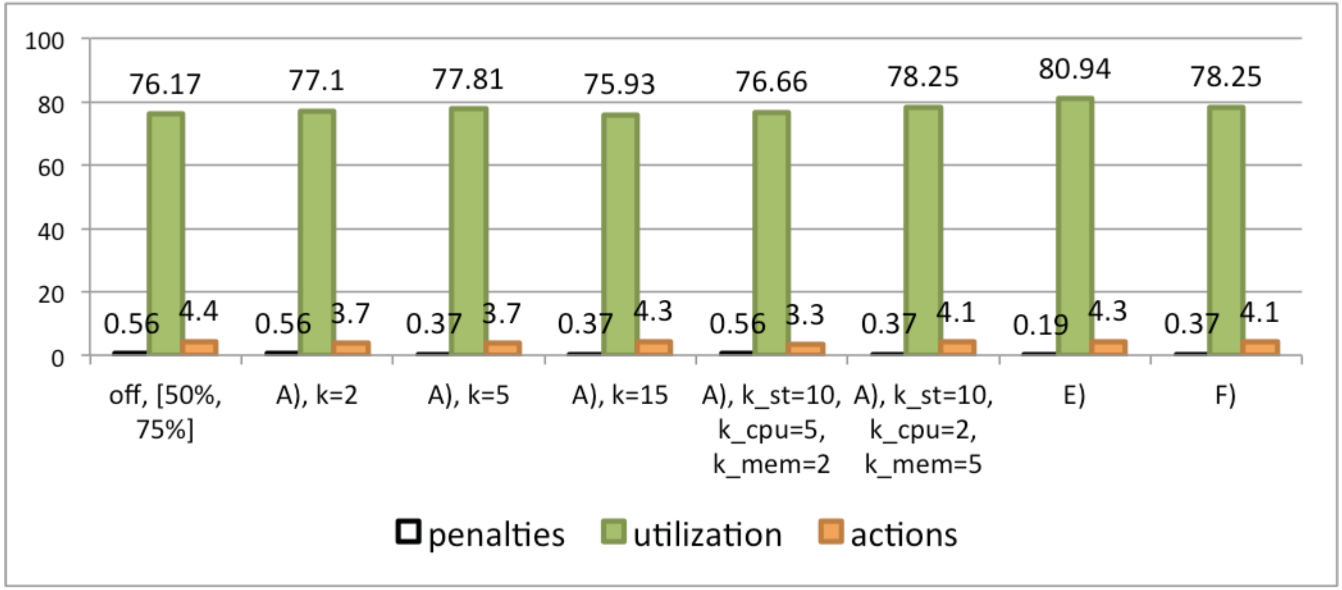
- Three scenarios
  1. Static configuration
  2. Speculative approach
  3. “Oracle Approach” - Peak provisioning - we know everything just before it happens

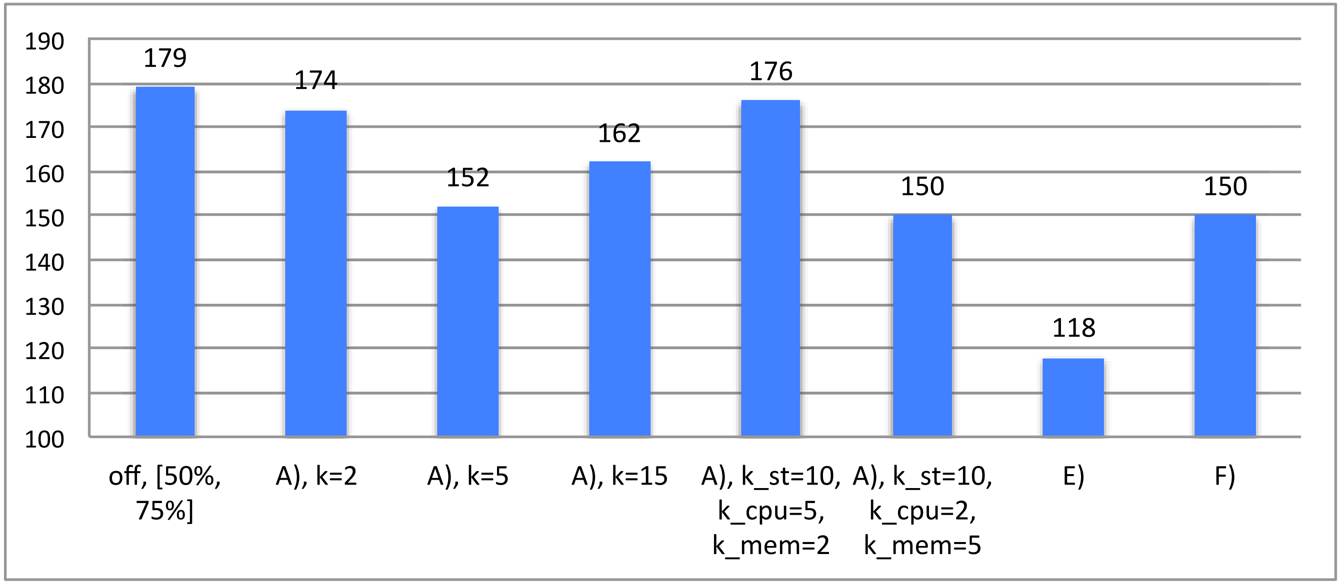




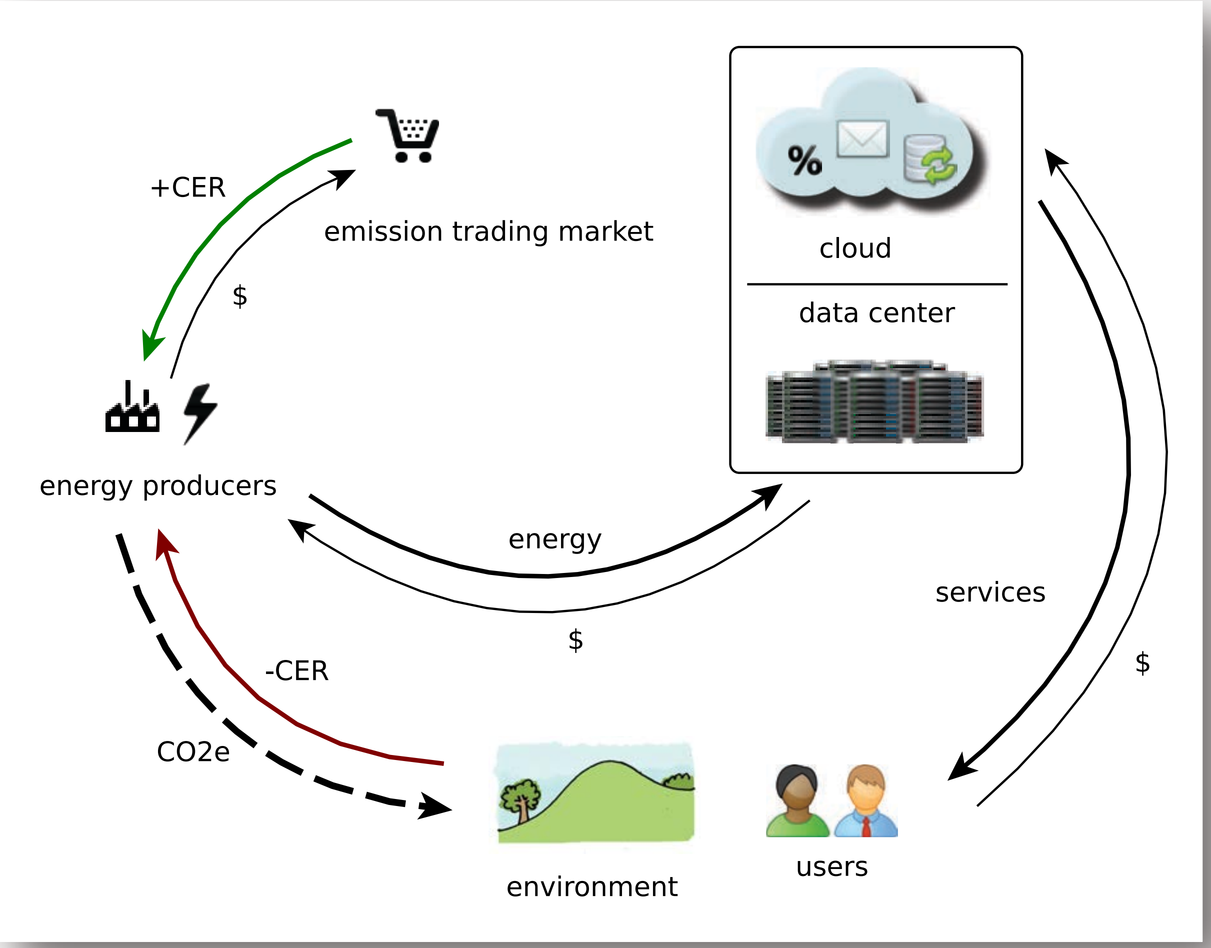


Bioinformatics Workflow



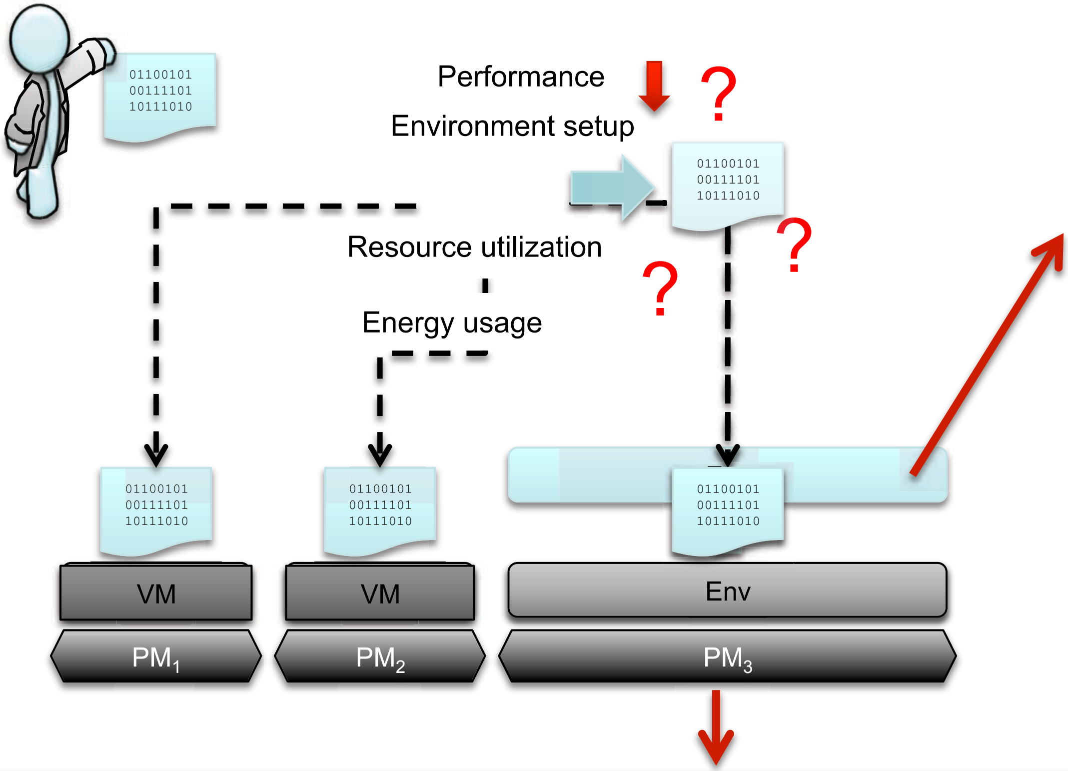


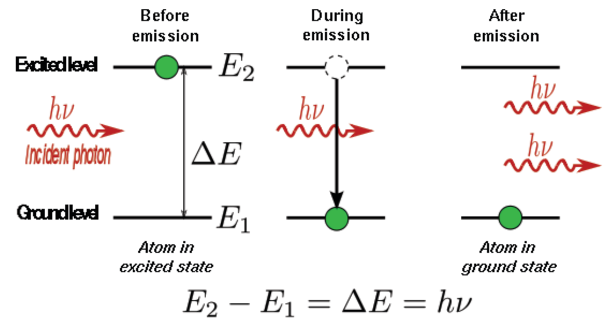
Future / Current Work: CO_2 trading

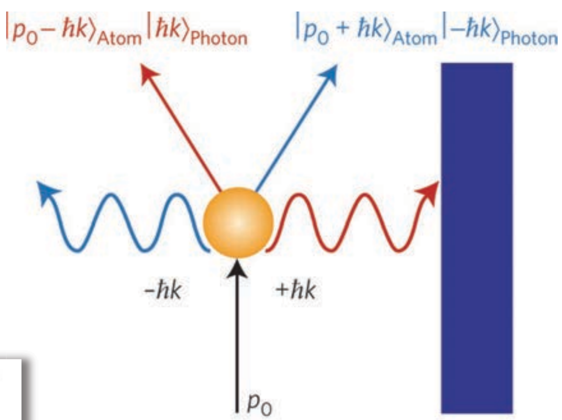


certified emission reductions (CERs)

Current / Future Work: Costs of moving to Clouds?







162	Vienna Scientific Cluster Austria	VSC-2 - Megware Saxonid 6100, Opteron 8C 2.2 GHz, Infiniband QDR Megware	20776	152.9	182.8	430

Cooperation with physics department (TU Vienna):
Application for yocto second photon emission from quark gluon plasmas

Current / Future Work

- Setup steps:

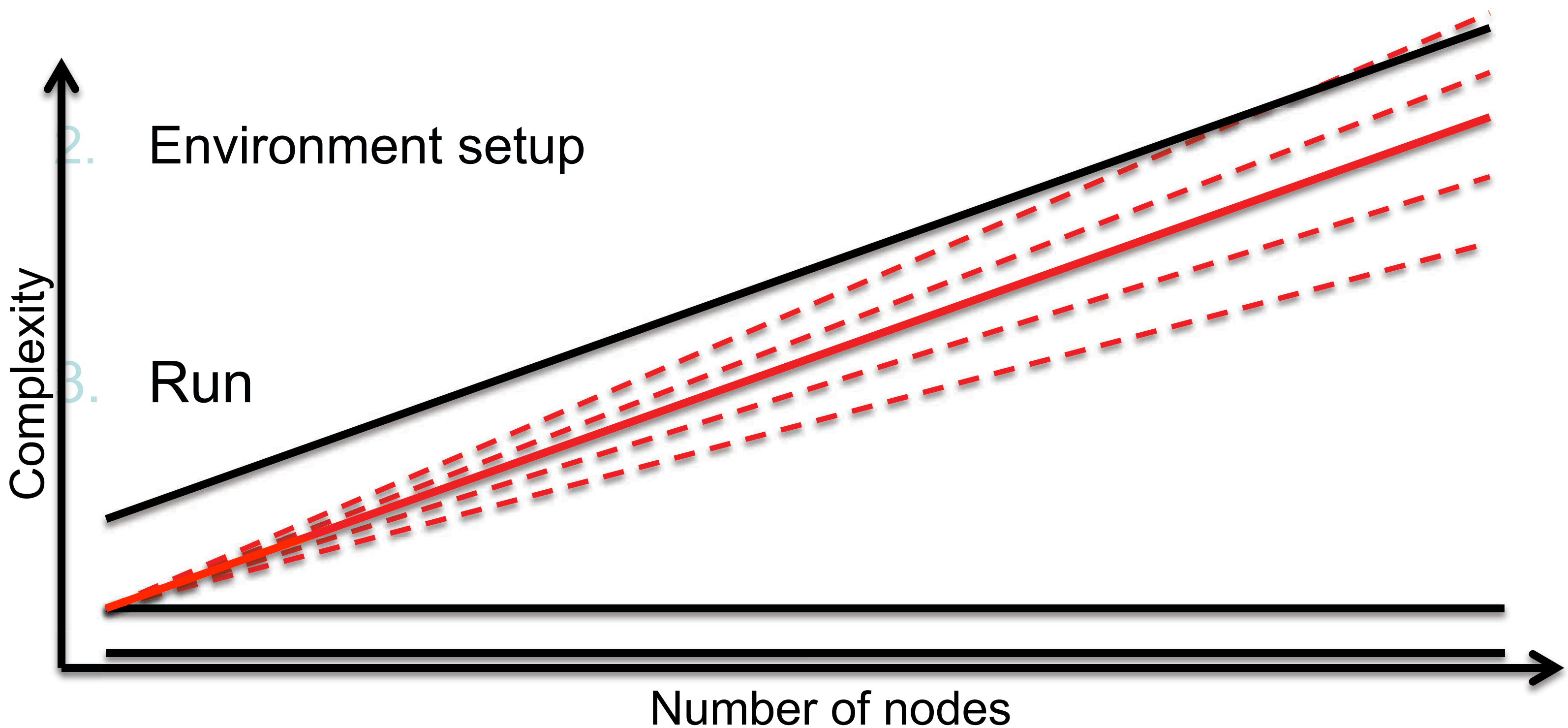
	Physical	Cloud
1. Platform setup	n	n
Installing an operating system	n	n
Installing drivers	n	n
Installing a hypervisor		n
Installing CMS		1
2. Environment setup	n	1
Installing libraries	n	VM image 1
Installing application	n	1
3. Run	1	1
Deploy		1
Run	1	1

Complexity model

* Setup steps:
  1. Platform setup
  2. Environment setup
  3. Run

- Three scenarios:

n	n	Initial setup
n	1	Next setup
1	1	Migration



Summary and Current / Future Work

- Work done
  - Self-adaptable Cloud Infrastructures
    - Generic Monitoring
    - Knowledge Management
    - Applied to real world scenarios (Bioinformatics)
- Future Work:
  - Systematic Approaches for the Development of Cloud Infrastructures
    - Software Engineering, Model Checking, Verification, Formal Models, ...
  - Big Data & Energy Efficiency
    - Data is the "new petrol" in Europe
    - Management of CO_2 footprints

Thanks to my colleagues

Soodeh Farokhi

Michael Maurer

Ivan Brešković

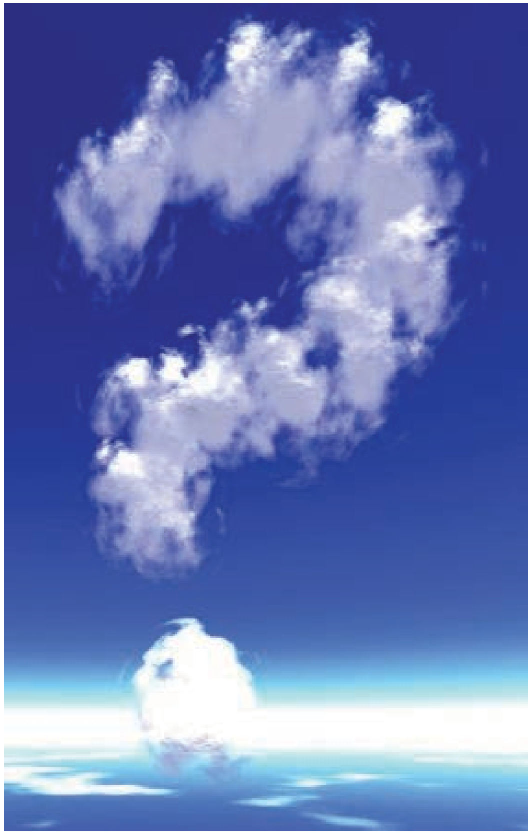
Andres Garcia Garcia

Dražen Lučanin

Toni Mastelić

Vincent Emeakaroha

Questions & Contact information



Ivona Brandic

Distributed Systems Group

Information Systems Institute

Vienna University of Technology

Austria

email: ivona@infosys.tuwien.ac.at

http://www.infosys.tuwien.ac.at/staff/ivona