

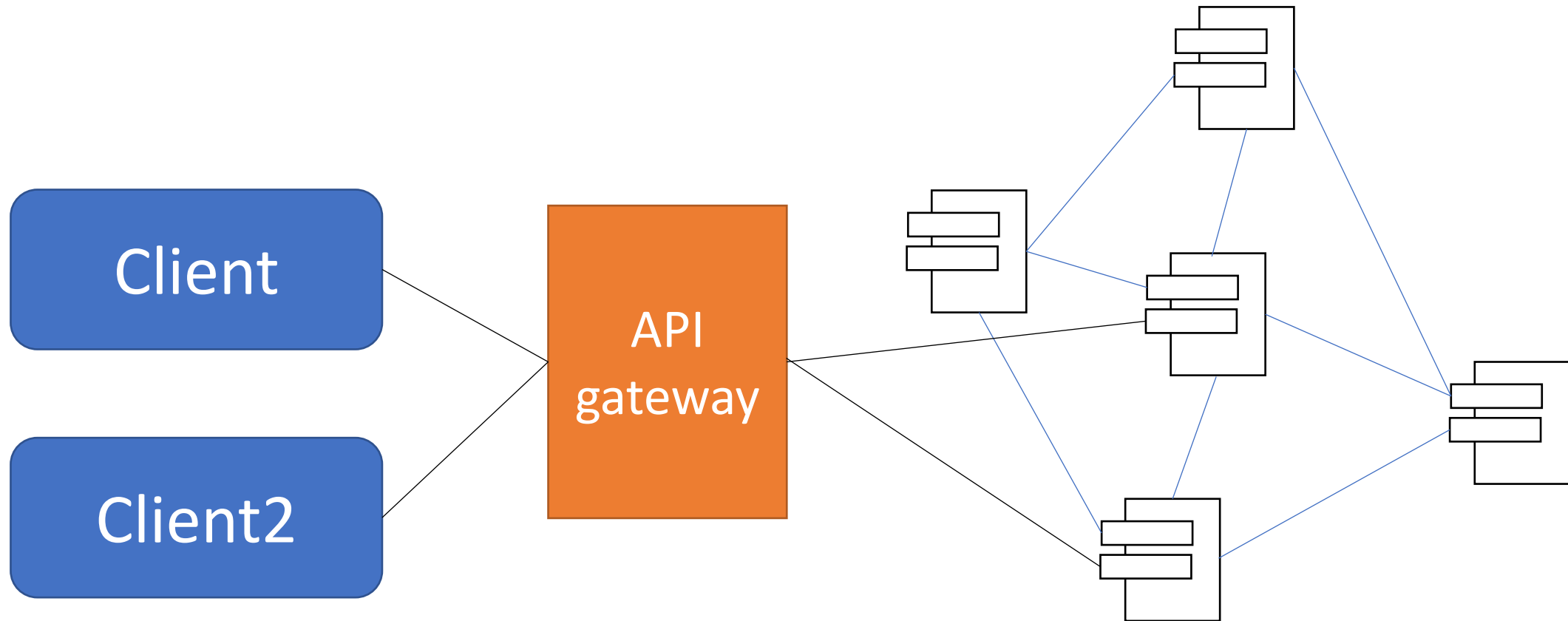


Hosting&monitoring

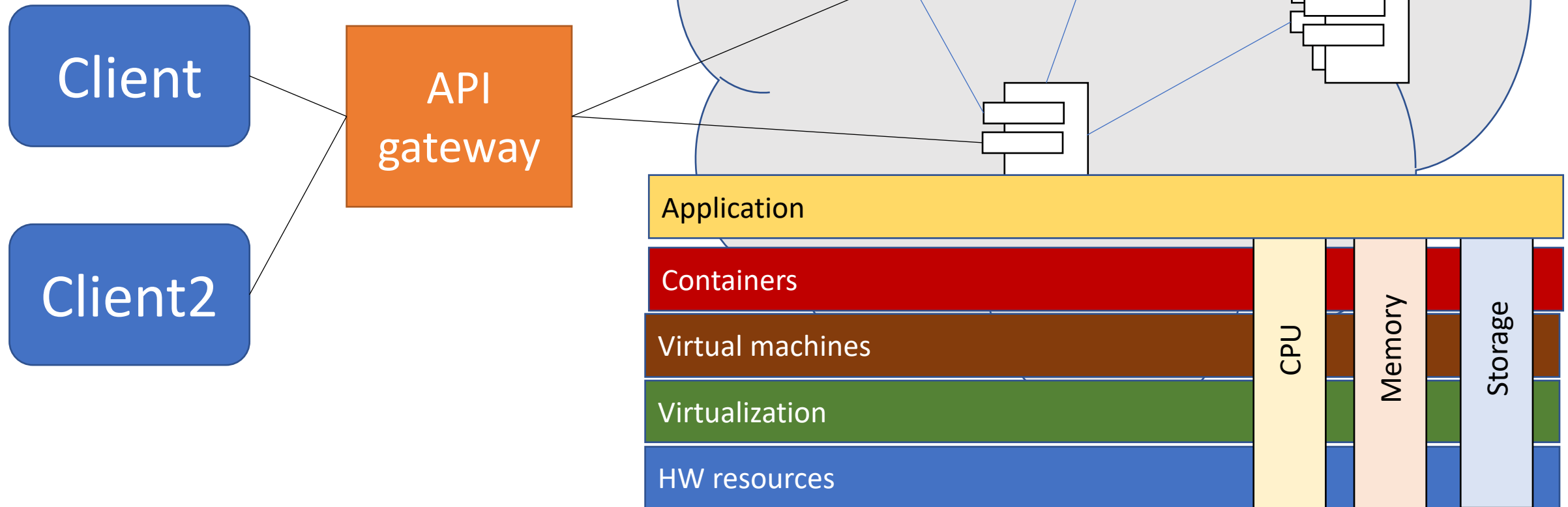
Kari Systä

How to monitor

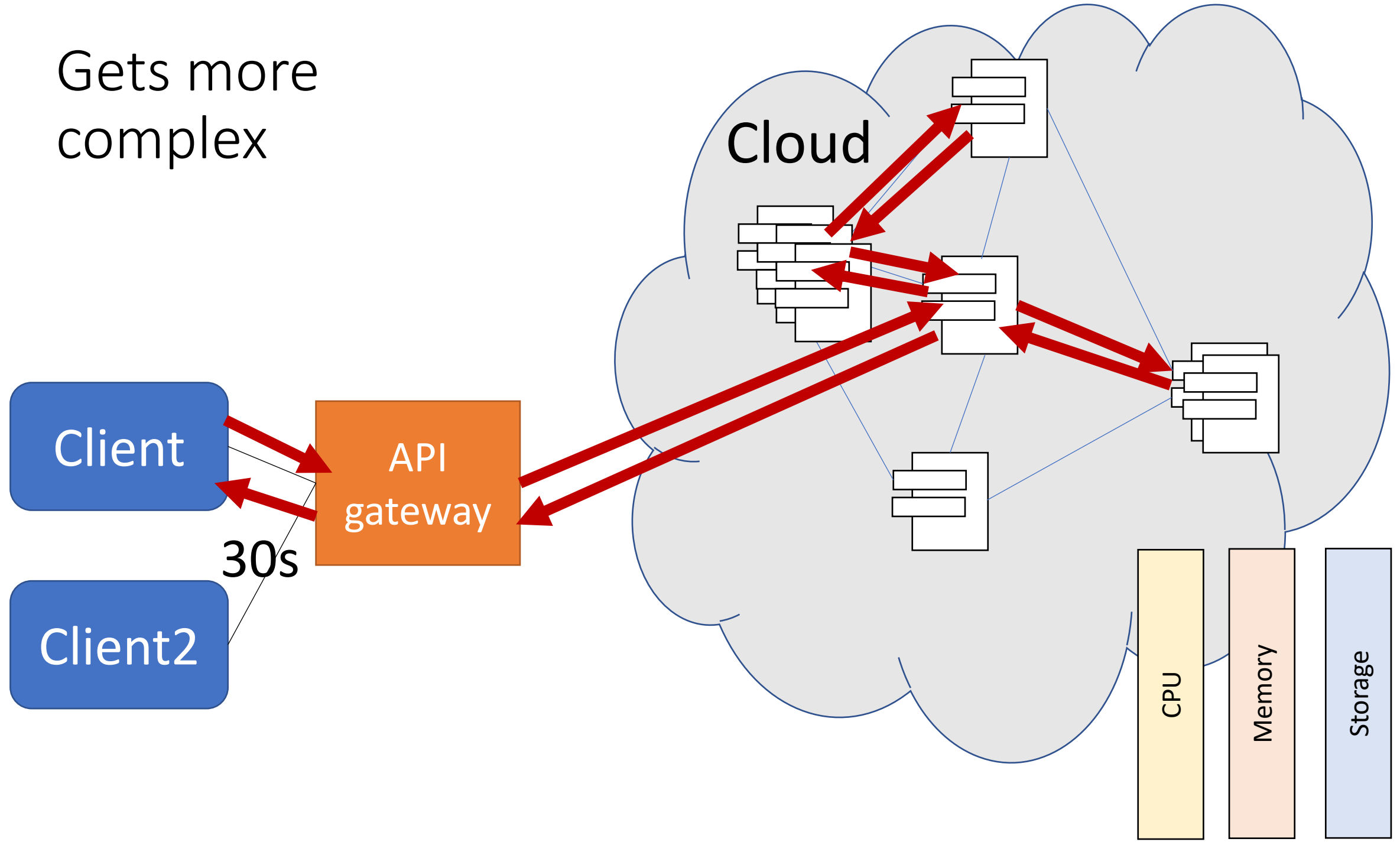
Recall a possible microservice architecture



Gets complex



Gets more complex



Is the problem

- In your application code?
- In your infrastructure code?

Example: Amazon CloudWatch

The screenshot displays the Amazon CloudWatch console interface. On the left, a navigation sidebar includes links for CloudWatch, Dashboards, Alarms (with a sub-menu for ALARM, INSUFFICIENT, and OK), Billing, Events, Rules, Logs, and Metrics. The main content area shows a list of alarms filtered by 'State is OK'. One alarm, 'SSM CPU Alarm', is selected and its details are shown below. The alarm's state is 'OK', and its configuration includes a threshold of 'CPUUtilization >= 30 for 3 minutes'. The 'State Details' section explains that the state changed to OK because the threshold was crossed by a single data point that was not greater than or equal to the threshold. A graph on the right visualizes the CPU utilization over time, showing a red horizontal line at the 30% threshold and a blue area representing the utilization, which peaks above the threshold around 04:00 on 4/10.

State	Name	Threshold	Config Status
OK	SSM CPU Alarm	CPUUtilization >= 30 for 3 minutes	

Alarm:SSM CPU Alarm

Details | History

State Details: State changed to OK at 2017/04/09. Reason: Threshold Crossed: 1 datapoint (0.16599999999999998) was not greater than or equal to the threshold (30.0).

Description: EC2 Instance Alarm based on CPU Utilization

Threshold: CPUUtilization >= 30 for 3 minutes

Actions: In ALARM: • Send message to topic "awsconfig-topic"
Send message to topic "awsconfig-topic" (shashikp@amazon.com)

Namespace: AWS/EC2

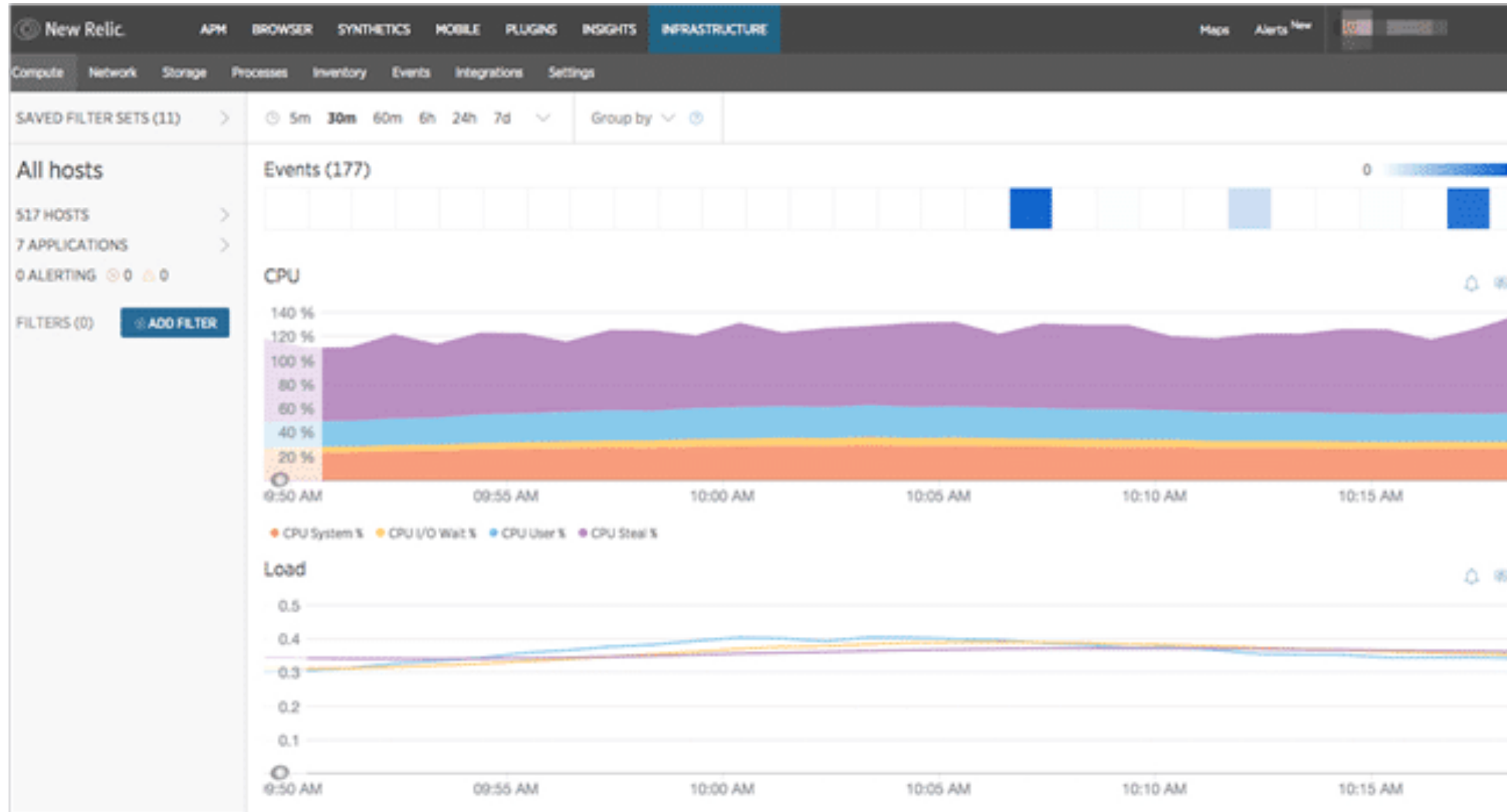
Metric Name: CPUUtilization

Dimensions: InstanceId = i-069b170e1098099df (ssm-2)

SSM CPU Alarm
CPUUtilization >= 30

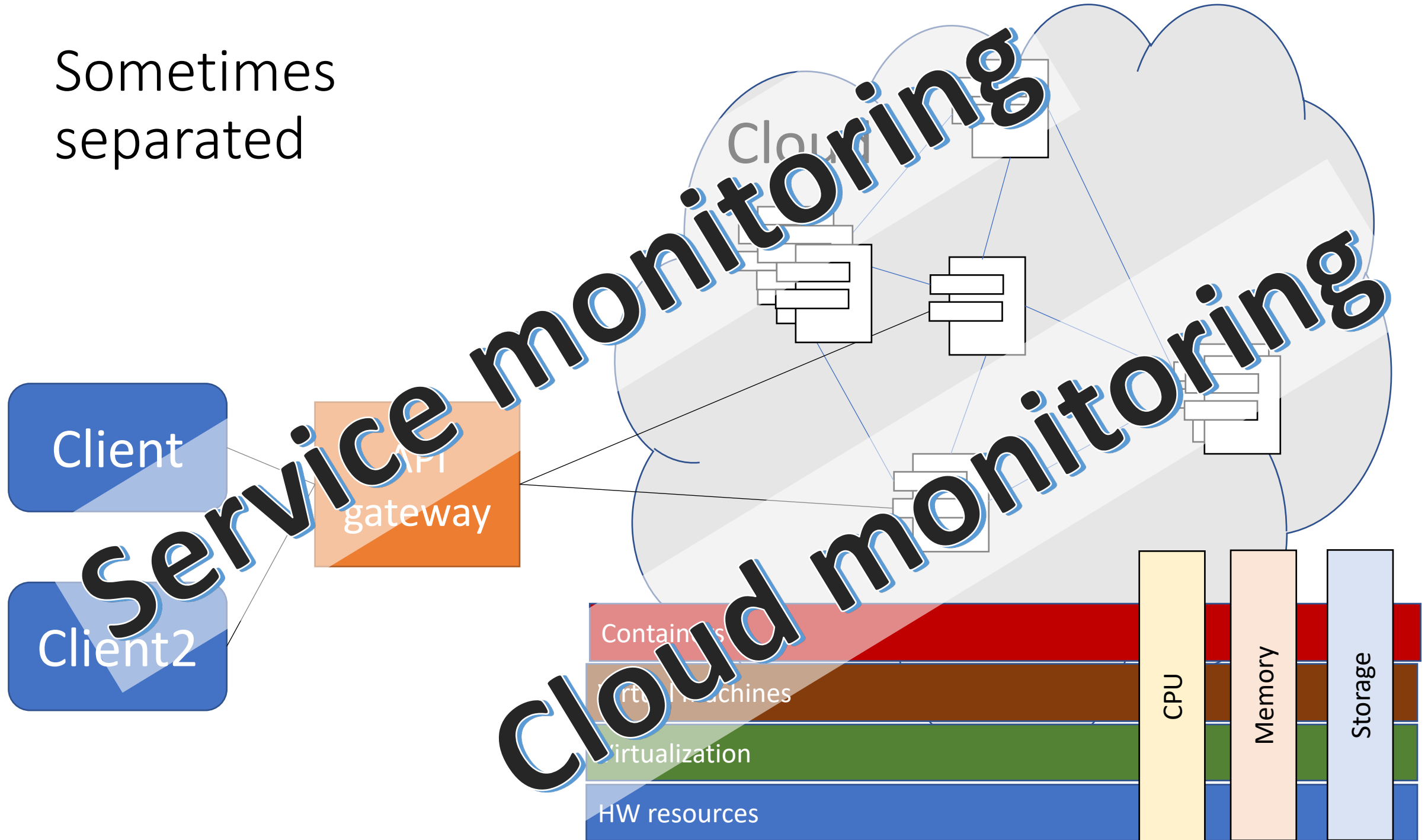
Graph showing CPU Utilization over time (4/10 02:00 to 04:00). The threshold is 30.0. The utilization spikes above the threshold around 04:00.

Example: New Relic





Sometimes separated



What might be monitored

- Availability
- MTBF (mean time between failures)
- Throughput
- Response time
- Latency
- Security threats
- Scalability
- Cost per customer
- Usage (recall A/B testing)
- Application specific measures

In your project

- *(Optional)* implement monitoring and logging for troubleshooting. This should be a separate service that the user can use through browser. It should show at least start time of the service, number of requests it has received after start.
- **Waiting for creative solutions !**

Couple of cloud quality "terms"

- QoS (Quality of Service): measure of capacity, performance etc.
- SLA (Service Level Agreement): an agreement between provider client about capacity, performance etc.
 - Or at least promise

<https://www.wired.com/insights/2011/12/service-level-agreements-in-the-cloud-who-cares/>

(Thomas J. Trappler; “If It’s in the Cloud, Get it on Paper: Cloud Computing Contract Issues”

<https://er.educause.edu/articles/2010/6/if-its-in-the-cloud-get-it-on-paper-cloud-computing-contract-issues.>)

- Codifies the specific parameters and minimum levels required for each element of the service, as well as remedies for failure to meet those requirements.
- Affirms your institution’s ownership of its data stored on the service provider’s system, and specifies your rights to get it back.
- Details the system infrastructure and security standards to be maintained by the service provider, along with your rights to audit their compliance.
- Specifies your rights and cost to continue and discontinue using the service.

- Availability (e.g. 99.99% during work days, 99.9% for nights/weekends)
- Performance (e.g. maximum response times)
- Security / privacy of the data (e.g. encrypting all stored and transmitted data)
- Disaster Recovery expectations (e.g. worse case recovery commitment)
- Location of the data (e.g. consistent with local legislation)
- Access to the data (e.g. data retrievable from provider in readable format)
- Portability of the data (e.g. ability to move data to a different provider)
- Process to identify problems and resolution expectations (e.g. call center)
- Change Management process (e.g. changes – updates or new services)
- Dispute mediation process (e.g. escalation process, consequences)
- Exit Strategy with expectations on the provider to ensure smooth transition