Using a Lifecycle Model for Developing and Executing Real-Time Online Applications on Clouds

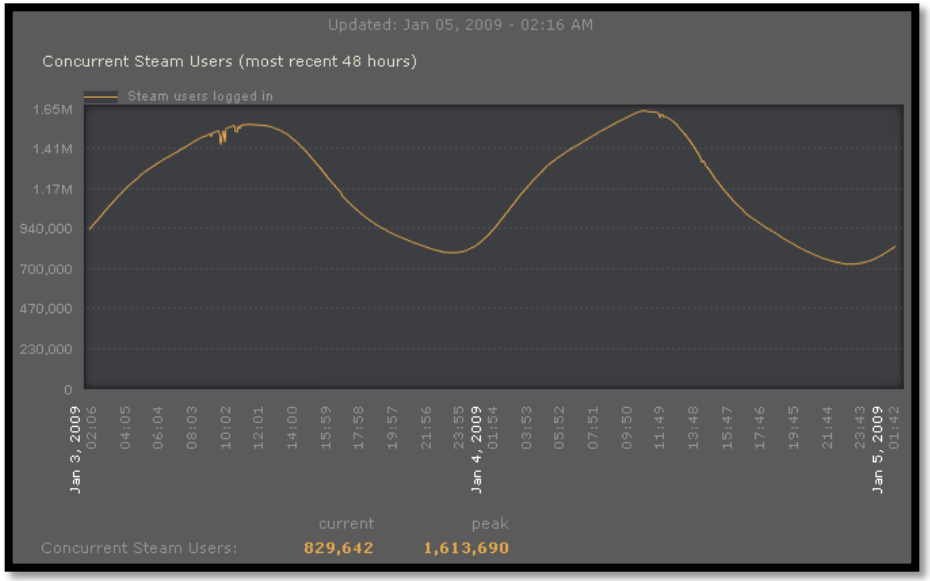
D. Meiländer, A. Bucchiarone, C. Cappiello, E. Di Nitto and S. Gorlatch

University of Muenster, Germany
Fondazione Bruno Kessler, Italy
Politecnico di Milano, Italy

Real-Time Online Interactive Apps (ROIA)

Real-Time Online Interactive Applications (ROIA), e.g., online games or interactive e-learning, are large-scale Internet apps with challenging requirements:

- Huge number of concurrent users in a single application instance (e.g., more than 40.000 simultaneous participants in Eve Online)
- Very high update rate of the app state (up to 100 Hz)
- Short response time to user actions (0.1 – 1.5 ms)
- Variable user load, daytime-dependent



Benefits of Clouds for ROIA Provision

Goal: Address these requirements by using Clouds/Infrastructure-as-a-Service (IaaS)

Benefits of Clouds/IaaS for ROIA:

- Choosing optimal hardware for particular ROIA
- Distribution of ROIA processing on arbitrary number of resources
- Dynamic adaptation of ROIA sessions
- Efficient and economic resource usage

State-of-the-art:

- Cloud platforms offer adaptation services based on generic system information (e.g., CPU load, bandwidth, etc.)
- **Open challenge:** Adaptation mechanisms taking into account application-specific monitoring information & real-time communication QoS

Our contributions:

- S-Cube Lifecycle Model for ROIA development and execution on Clouds addressing QoS aspects of ROIA and specific challenges on Cloud
  - startup times, static leasing periods, etc.
- Resource management system RTF-RMS implementing ROIA adaptation along the S-Cube Lifecycle Model

The S-Cube Lifecycle Model

Lifecycle for service-oriented apps defined & refined in the S-Cube project

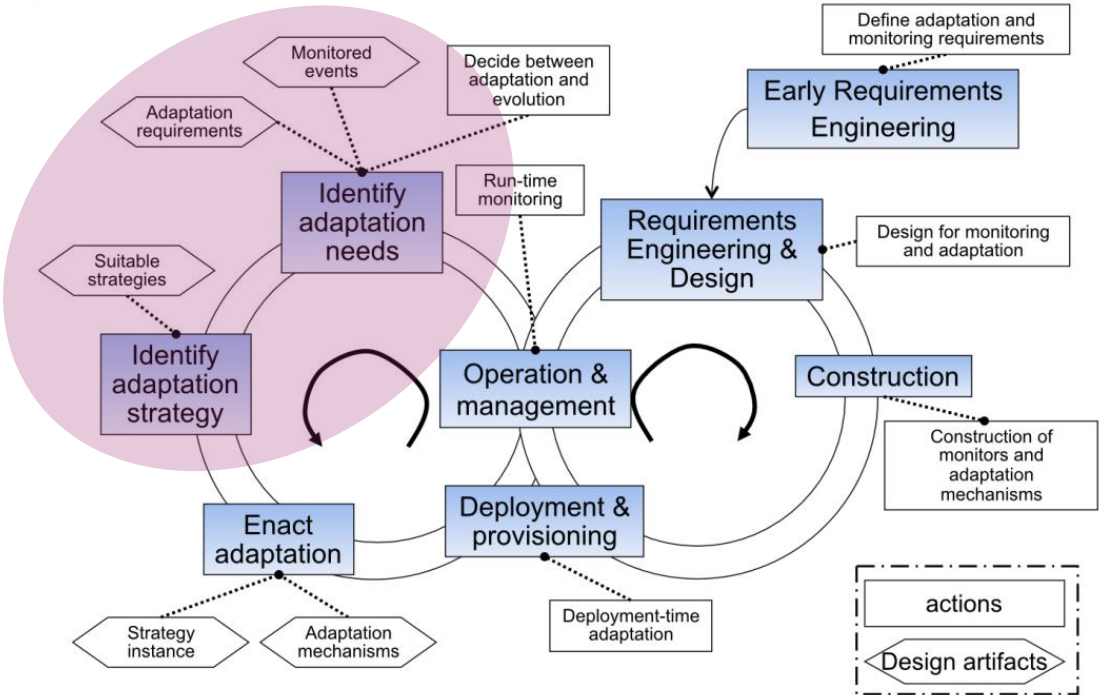
(1) A. Bucchiarone, C. Cappiello, E. Di Nitto, R. Kazhamiakin, V. Mazza, M. Pistore: Design for Adaptation of Service-Based Applications: Main Issues and Requirements, WESOA, 2009.

(2) S. Lane, A. Bucchiarone, I. Richardson, SOAdapt: A process reference model for developing adaptable service-based applications, Information and Software Technology, 2011.

Two coexisting cycles support each other during application lifetime

- **Evolution** (right cycle): Design-time iteration cycle, supporting long-term changes due to, e.g., changing user preferences, continuous adaptation actions
- **Adaptation** (left cycle): Runtime cycle addressing adaptation to support short-term changes, e.g., daytime-dependent user load

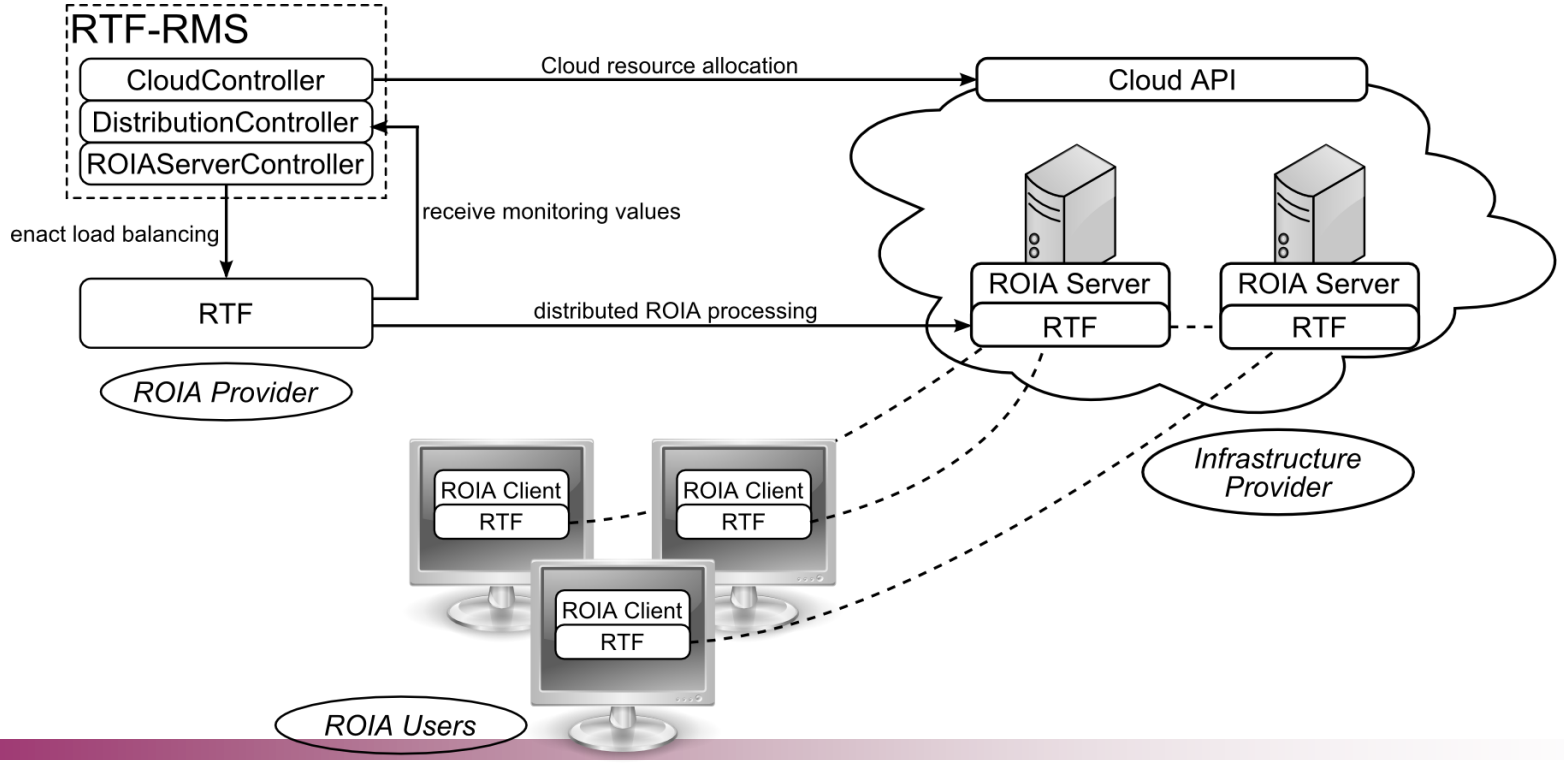
This talk: Adaptation triggers & strategies for ROIA execution on Clouds



The RTF-RMS Resource Mgt System

RTF-RMS implements adaptation according to the S-Cube Lifecycle Model:

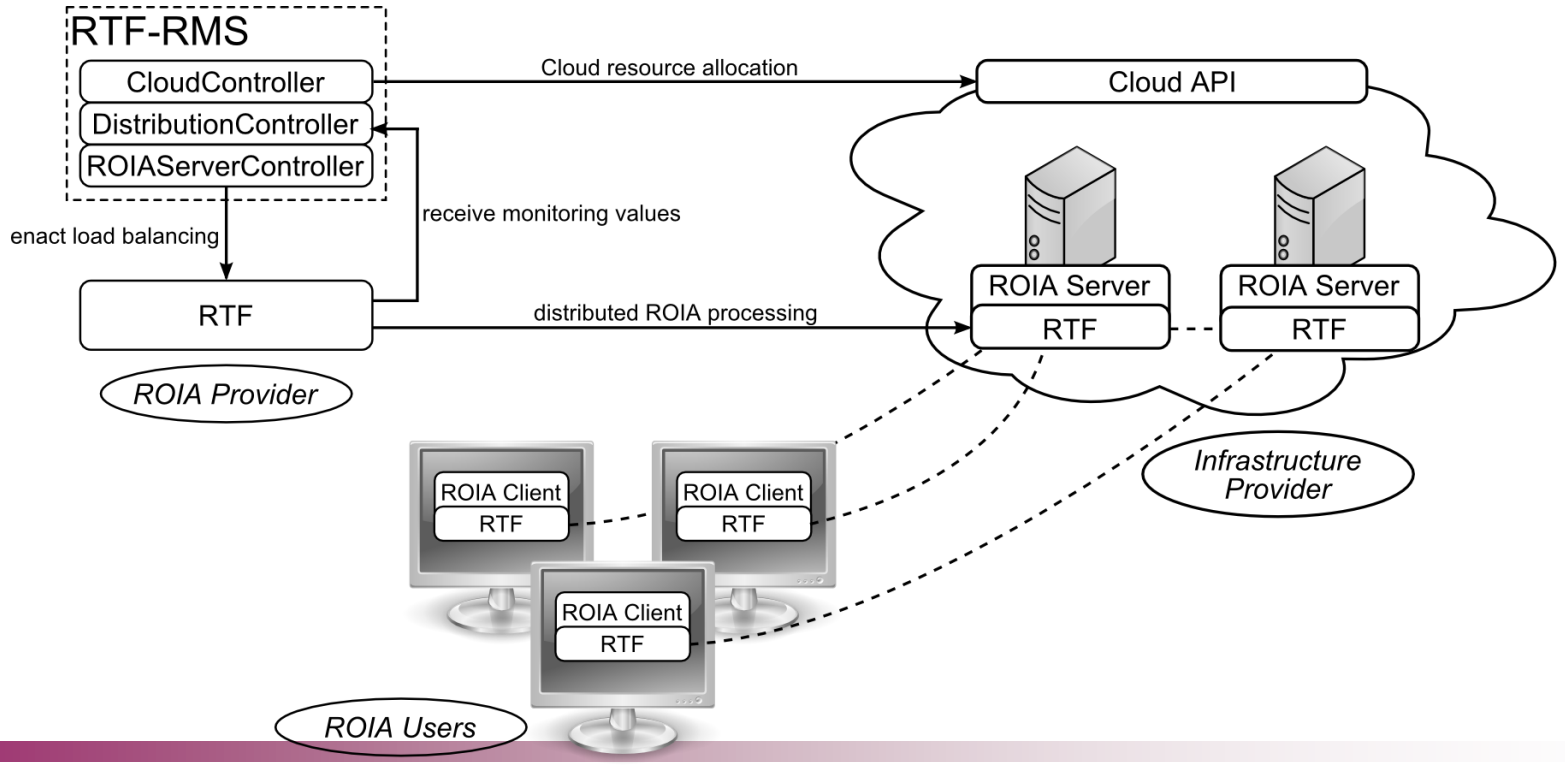
1. **Operation & Management:** receive monitoring values from RTF
2. **Identify adaptation need:** check monitoring data against predefined thresholds
3. **Identify adaptation strategy:** choose among four different load-balancing actions
4. **Enact adaptation:** enact the chosen load-balancing action using RTF



The RTF-RMS Resource Mgt System

RTF-RMS implements concrete adaptation mechanisms for ROIA:

- **Adaptation triggers** for ROIA based on application-specific monitoring, e.g.:
  - update rate, response time, etc.
- **Adaptation strategies** for distributed ROIA processing on Clouds



RTF-RMS: Adaptation Triggers

- Adaptation triggers must address app-specific requirements
  - e.g., game state updates must be computed faster for fast-paced action games than for role-playing games

- Adaptation triggers in RTF-RMS are configured by specifying an **application profile** that identifies:
  - relevant monitoring values
  - upper and lower thresholds for each value

```
<appProfile>
 <metric>
  <name>UpdateRate</name>
  <monitoringInterval>1</monitoringInterval>
  <addResourceThreshold>25</addResourceThreshold>
  <removeResourceThreshold>200</removeResourceThreshold>
 </metric>
</appProfile>
```

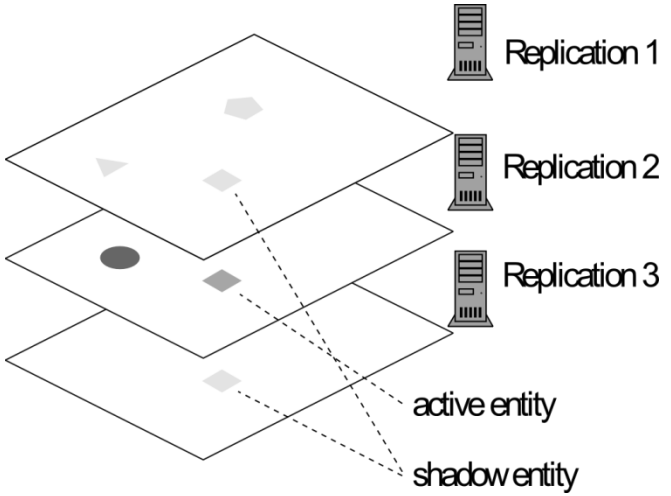
Example application profile

- Adaptation triggers must consider startup time of Cloud resources (up to several minutes)

→ RTF-RMS introduces a **resource buffer** to start a predefined number of resources in advance

RTF-RMS: Adaptation Strategies

- RTF-RMS chooses between the following four adaptation strategies:
  1. **User migration:** client connections are switched seamlessly between two application servers replicating the same zone
  2. **Replication enactment:** new servers are added to provide more computation power to a highly frequented zone by replicating it
  3. **Resource substitution:** running resources are substituted by more powerful resources
  4. **Remove resource:** dispensable resources are removed



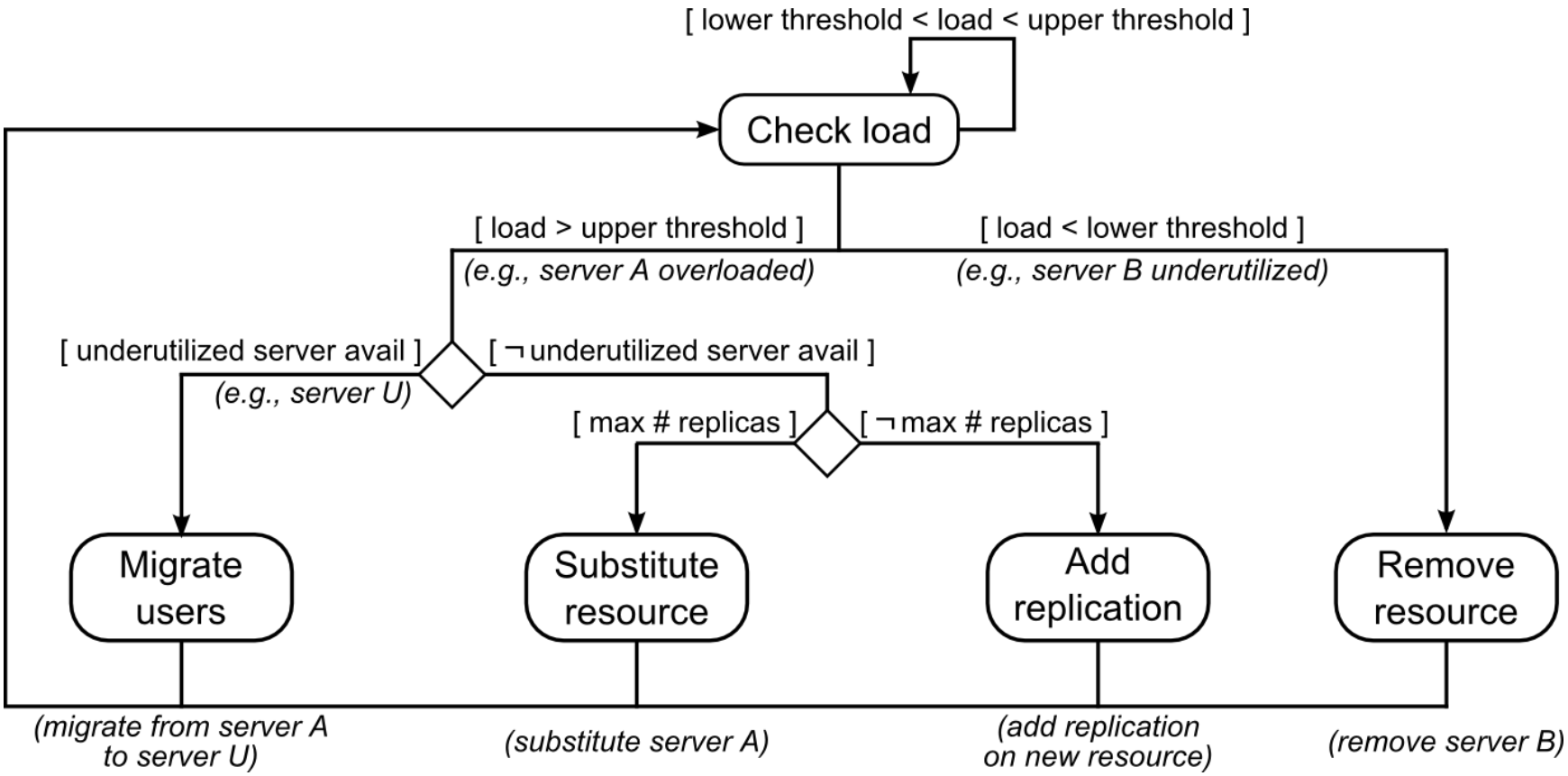
Replication: a single zone is processed by multiple application servers

- Adaptation strategies must consider static leasing periods, e.g., resources paid per hour

→ RTF-RMS stores dispensable resources in the resource buffer until they are demanded again or their leasing period ends

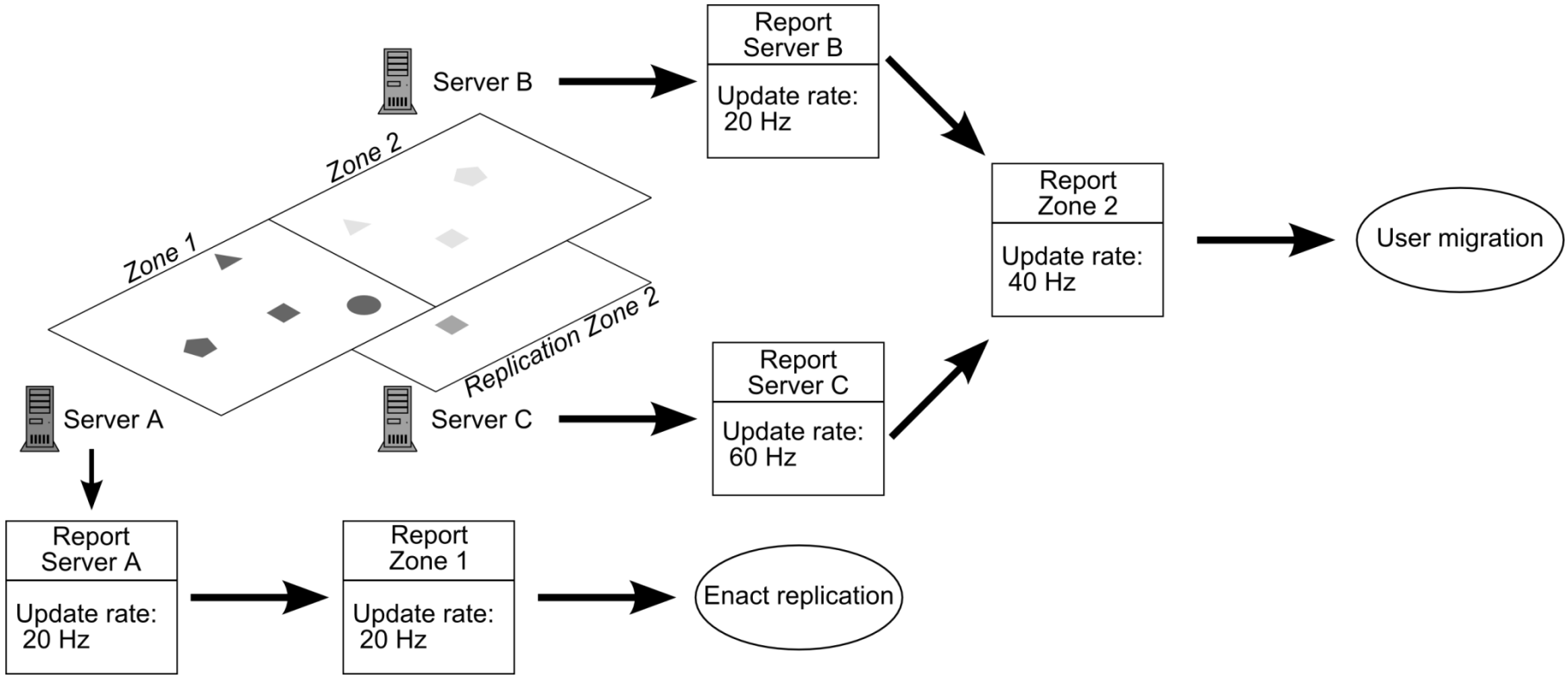
RTF-RMS: Identify Adaptation Strategy

An adaptation strategy is chosen on basis of three conditions as follows:



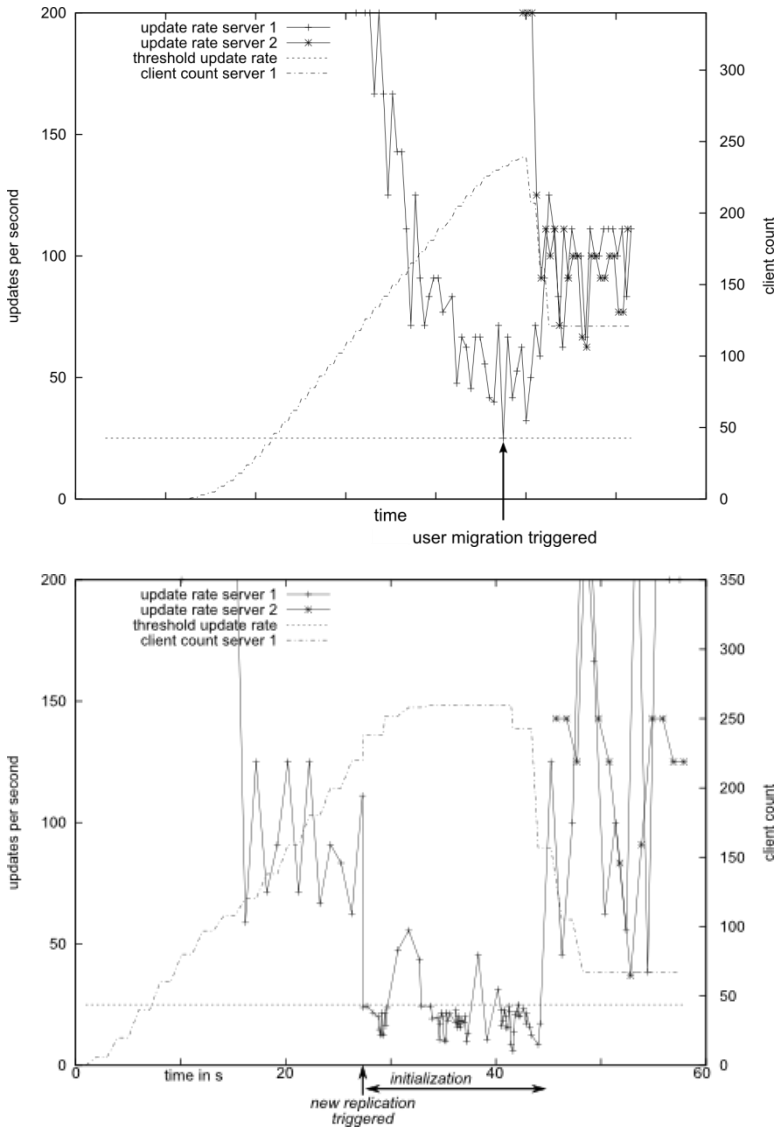
Identify Adaptation Strategy: Example

- Monitoring data is summarized in **zone reports**
- Average load is compared with thresholds
  - e.g., `addResourceThreshold` update rate: 25 Hz



Experiments

- **Experimental setup:**
  - Fast-paced 3D online game
  - 260 clients are simulated by bots moving randomly & shooting at opponents
  - update rate should be > 25 Hz
- **User migration (top picture):**
  - Avg. update rate increased from 25 Hz to ~100 Hz
- **Replication enactment (bottom picture):**
  - Integration in app processing in ~15 s
  - Without resource buffer: > 130 s
  - Avg. update rate increased from > 20 Hz to ~100 Hz
- **Result:** RTF-RMS adaptation improves QoS for a changing number of users



Contributions

- **Lifecycle Model** for development & execution of ROIA addressing specific challenges related to Clouds
  - Adaptation triggers based on app-specific monitoring instead of generic system information about resource utilization, e.g., Amazon CloudWatch
  - Four different adaptation strategies reflecting the interactivity between ROIA users: user migration, replication, resource substitution and resource removal

- **Resource management system** implementing adaptation
  - Suitable Cloud resource allocation for ROIA minimizing startup times & utilizing leasing periods using the resource buffer
  - Experimental results demonstrate successful & efficient load balancing

The research leading to these results has received funding from S-Cube, the European Network of Excellence in Software Services and Systems