Design by Units

Abstractions for Human and Compute Resources for Elastic Systems

Schahram Dustdar

Distributed Systems Group

TU Wien

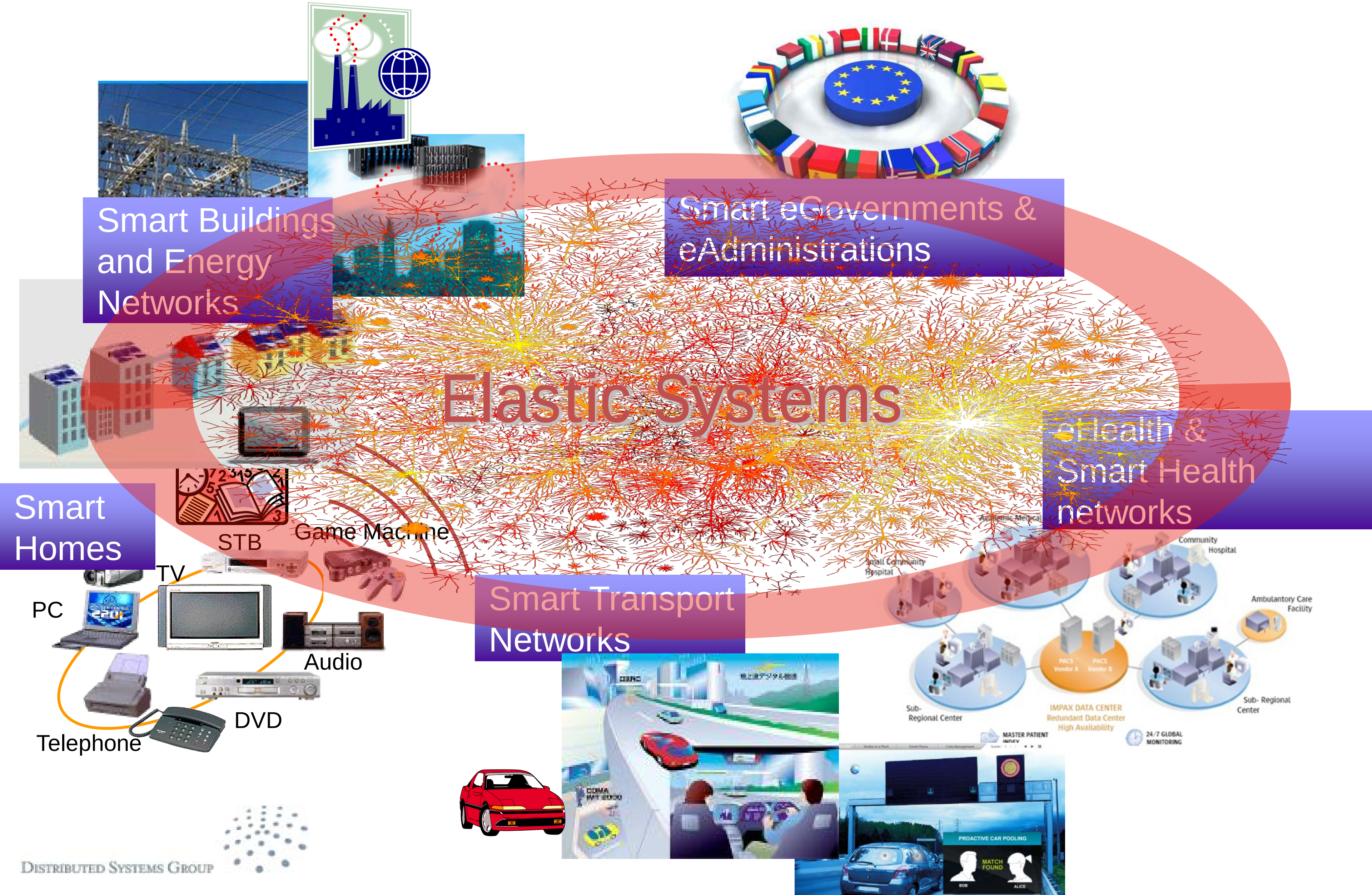
Distributed Systems Group

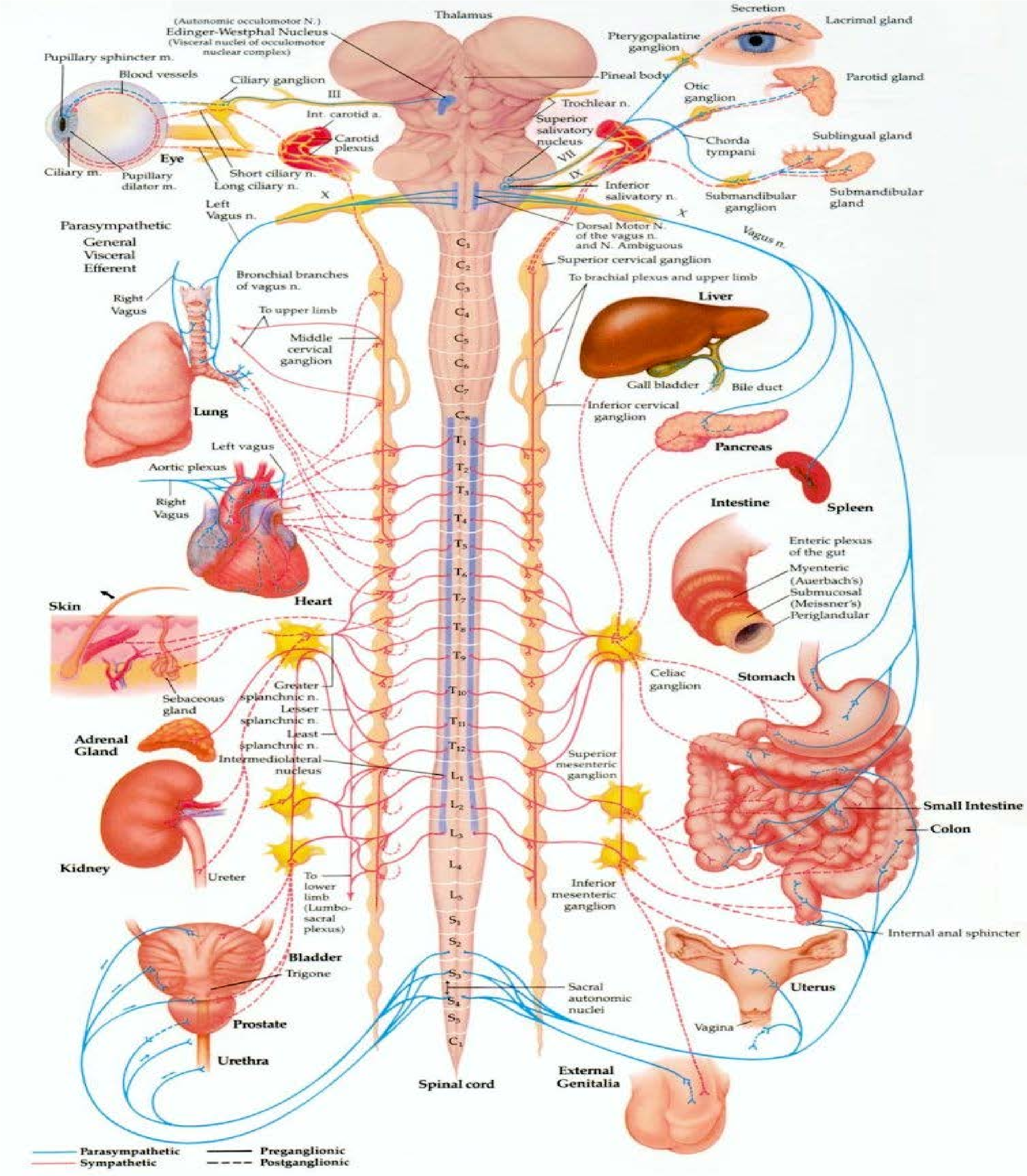
Structure

1. Understanding advanced Services Systems

2. Finding Abstractions/Programming model for Human-, Compute-, and Storage Resources -> *Design by Units*

Smart Evolution – People, Services, Things





Autonomic Nervous System

Managed City Governance Service Oriented Architecture


ICT for energy savings in buildings
Command Control Center
SMEs
Dashboards
User interfaces
Report
Reports
CO_2
Carbon footprint measurement
Benchmarking
Remote monitoring
Engineers
Villas
Fire
Safety & security
Energy
HVAC
CCTV
Carbon footprint
Factories
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
Schools
Fire
Safety & security
Energy
Chiller / HVAC
CCTV
Carbon footprint
Commercial & residential buildings
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
Utilities
Sewage pumps
Water treatment plants
Irrigation
Hospitals
Fire
Lift
Safety & security
Energy
Chiller / HVAC
Boiler
CCTV
Carbon footprint
© Copyright 2010 Pacific Control Systems. All Rights Reserved.

ICT enabled Security Services

ICT enabled Telematics

ICT enabled services for food storage and delivery

Cold storage system

Freezer rooms

Food display cabinets

ICT enabled services for health care

Hospital equipments monitoring

Hospital operations management

Hospital security systems

ICT enabled smart education systems

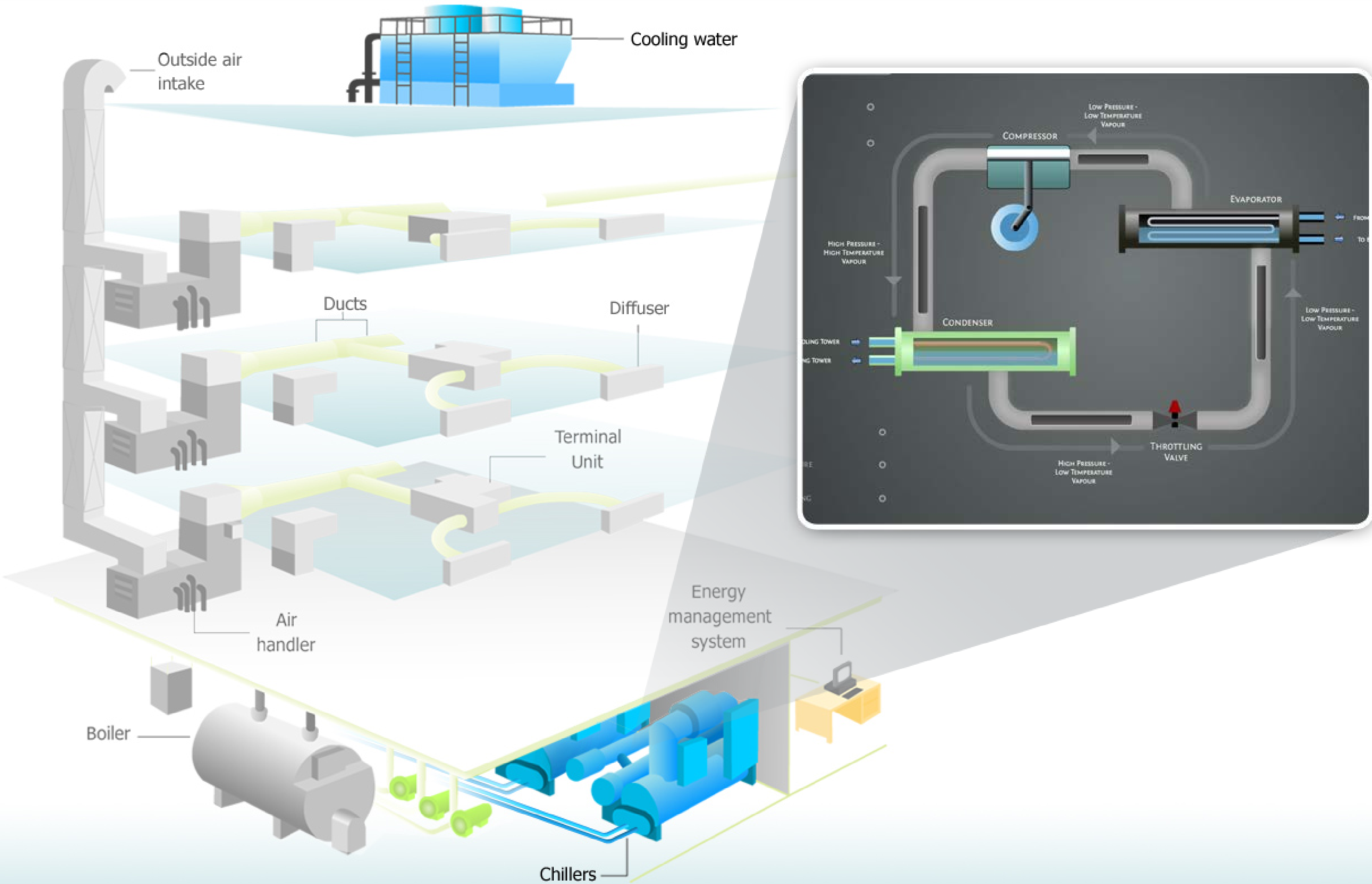
Smart classrooms

Smart Universities

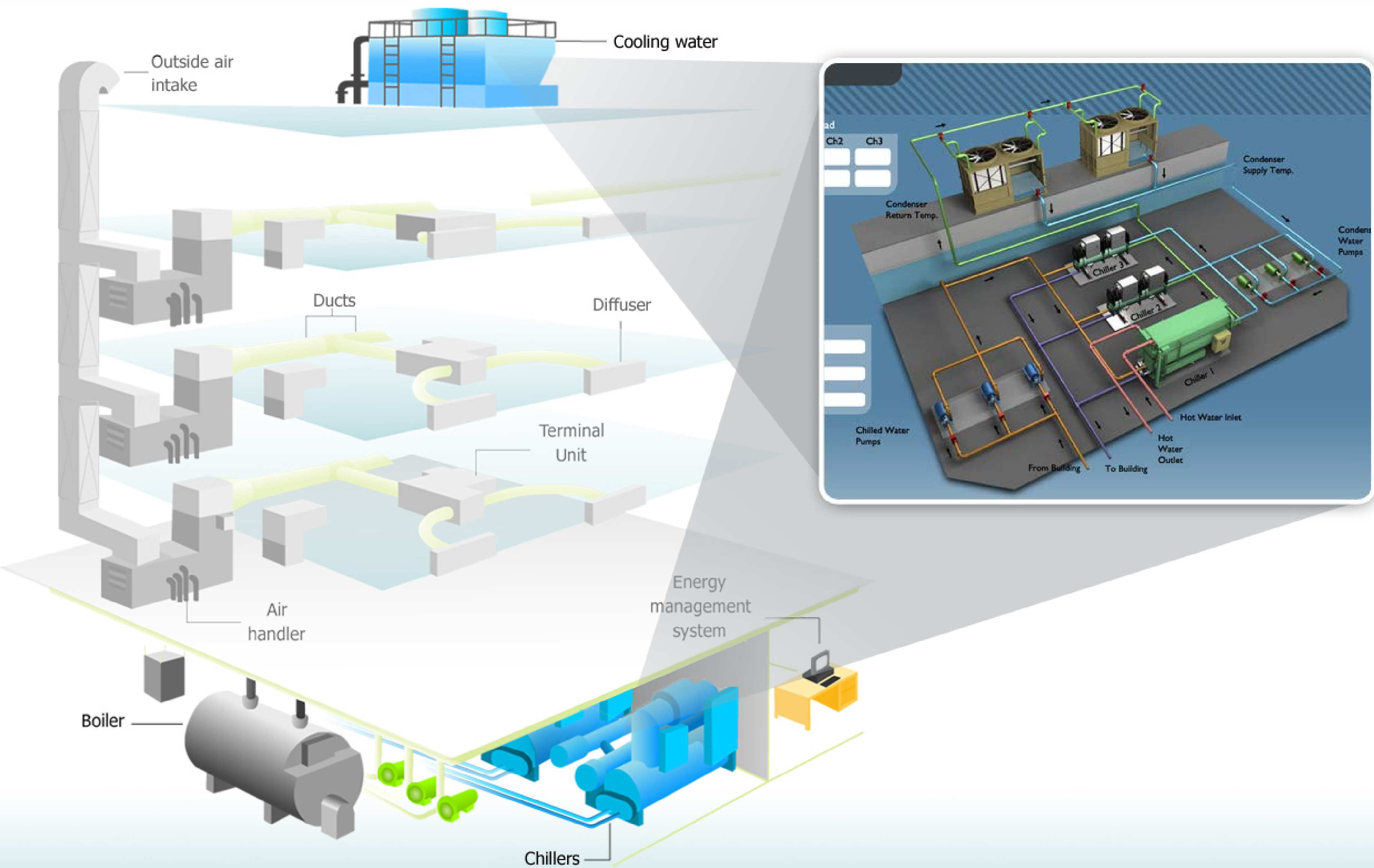
Campus infrastructure

Distributed Systems Group

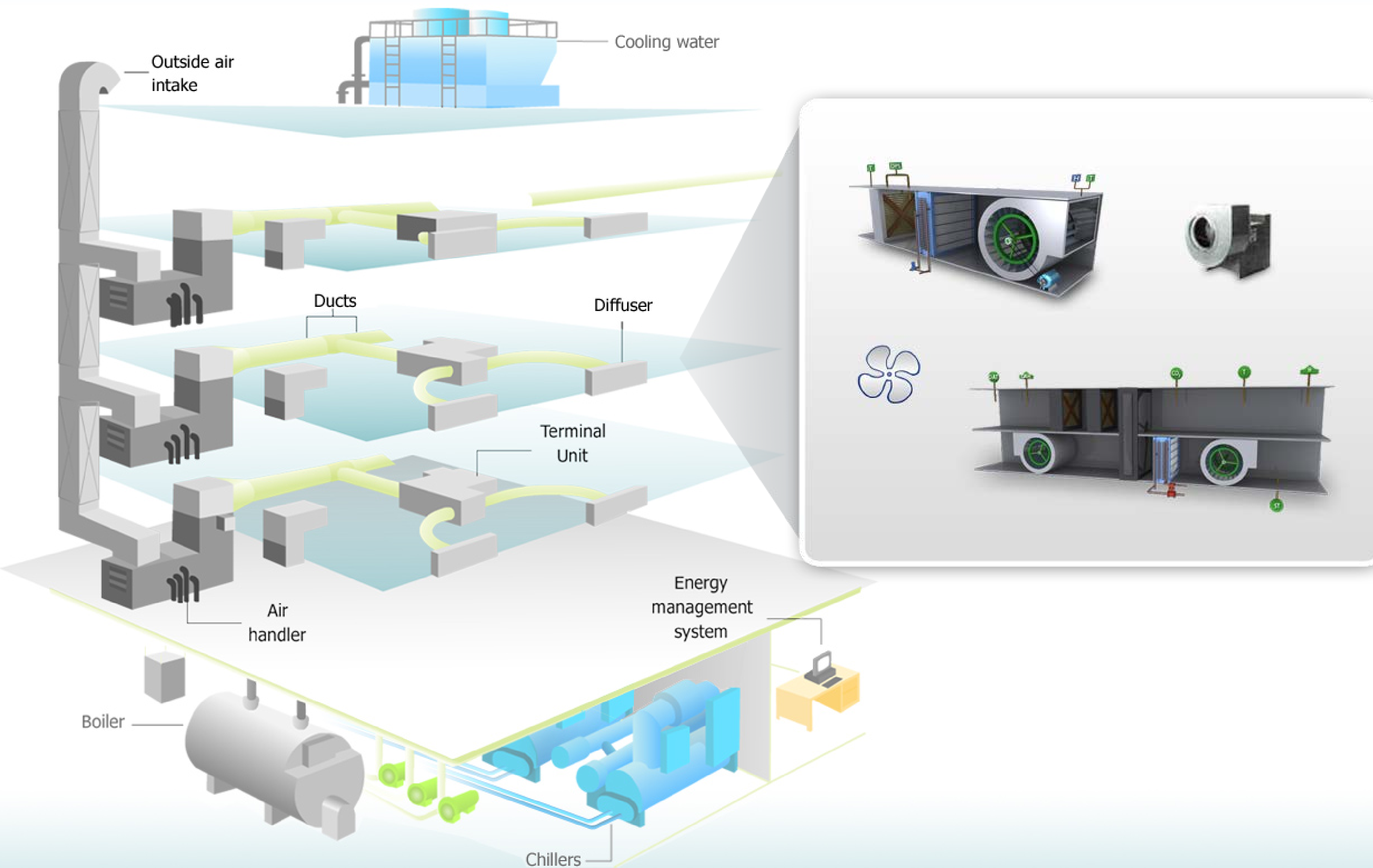
HVAC (Heating, Ventilation, Air Conditioning) Ecosystem



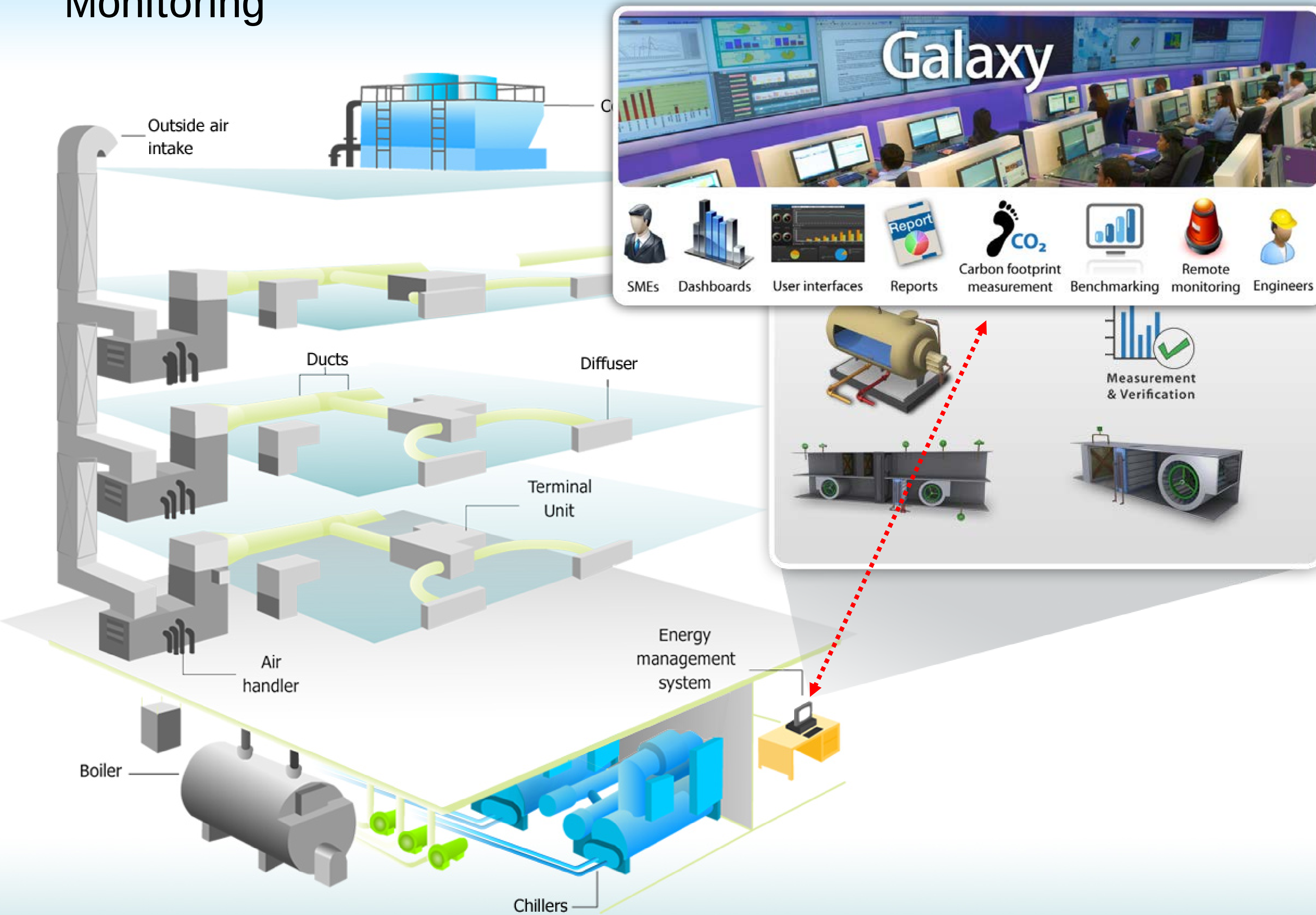
Water Ecosystem



Air Ecosystem

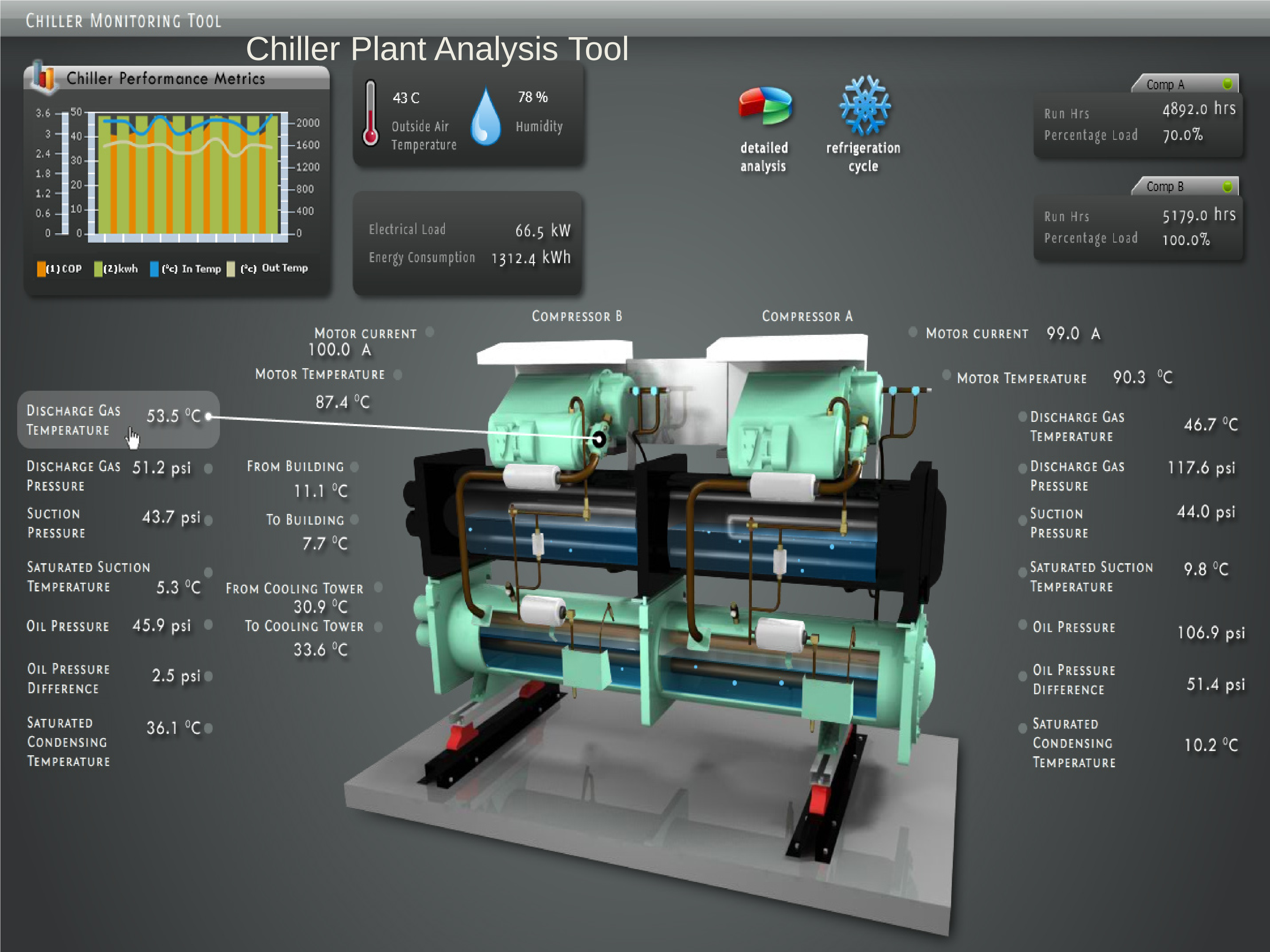


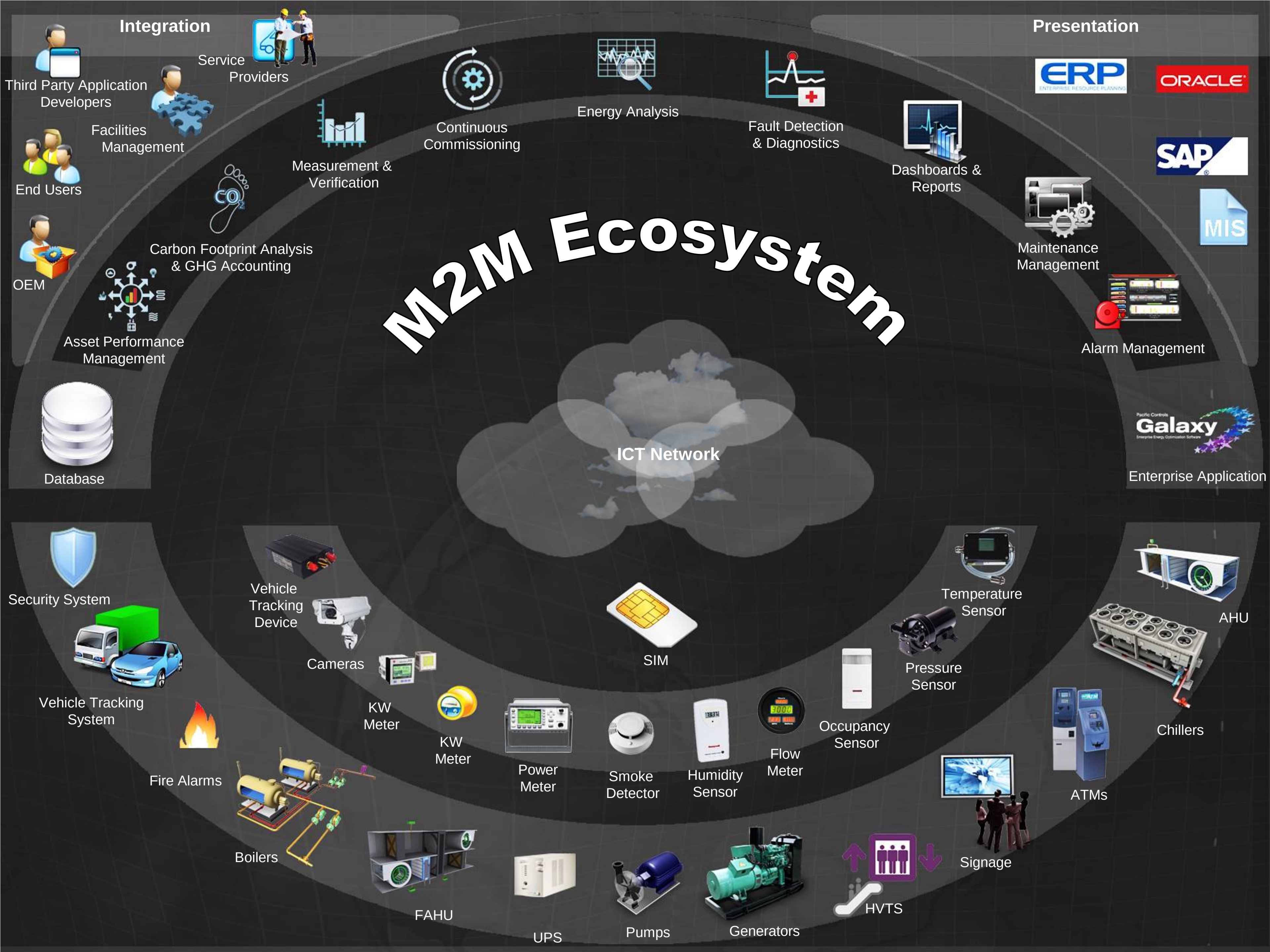
Monitoring



Galaxy
SMEs
Dashboards
User interfaces
Report
Reports
Carbon footprint measurement
Benchmarking
Remote monitoring
Engineers
Measurement & Verification
Outside air intake
Ducts
Diffuser
Terminal Unit
Air handler
Energy management system
Boiler
Chillers

Chiller Plant Analysis Tool

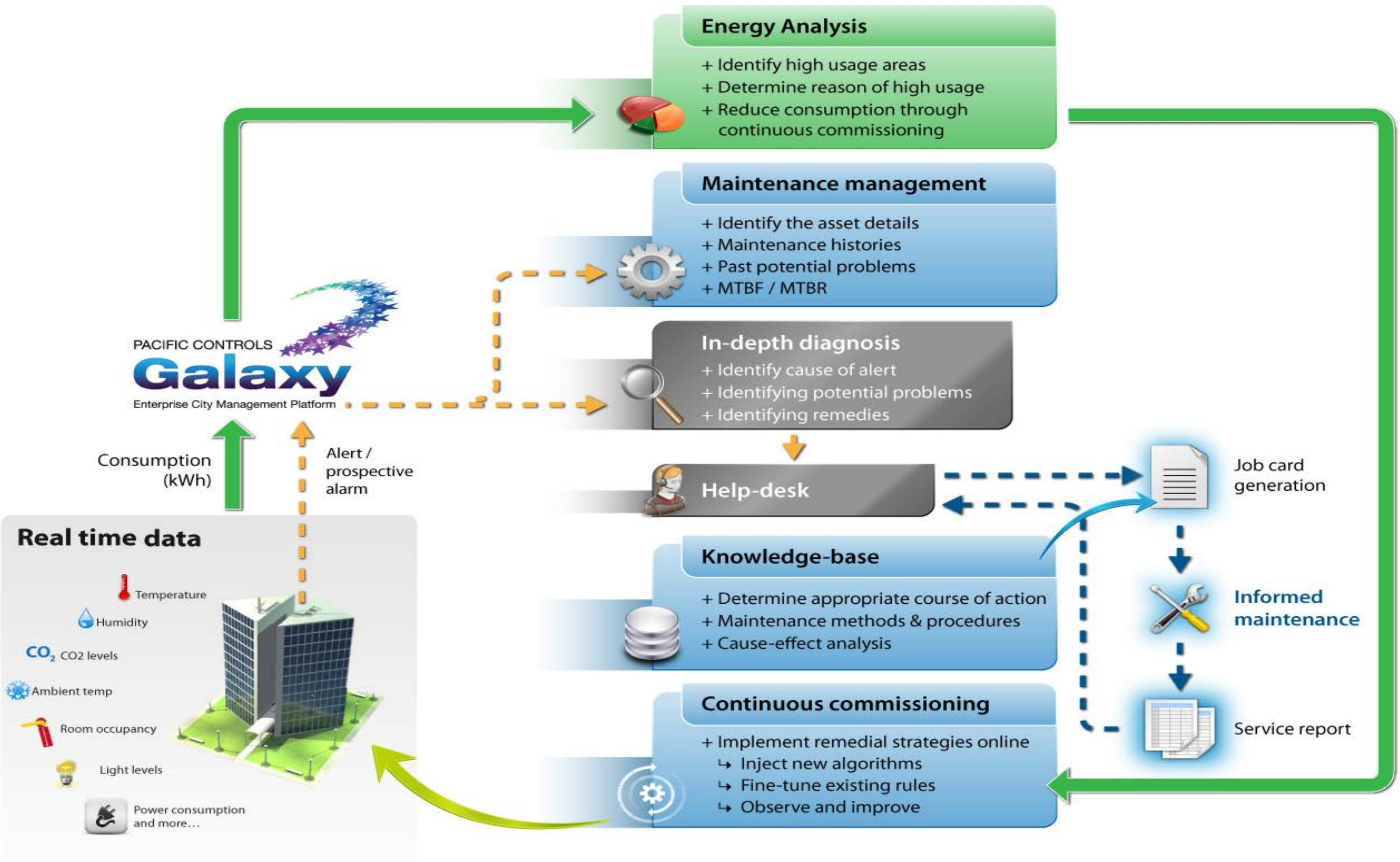




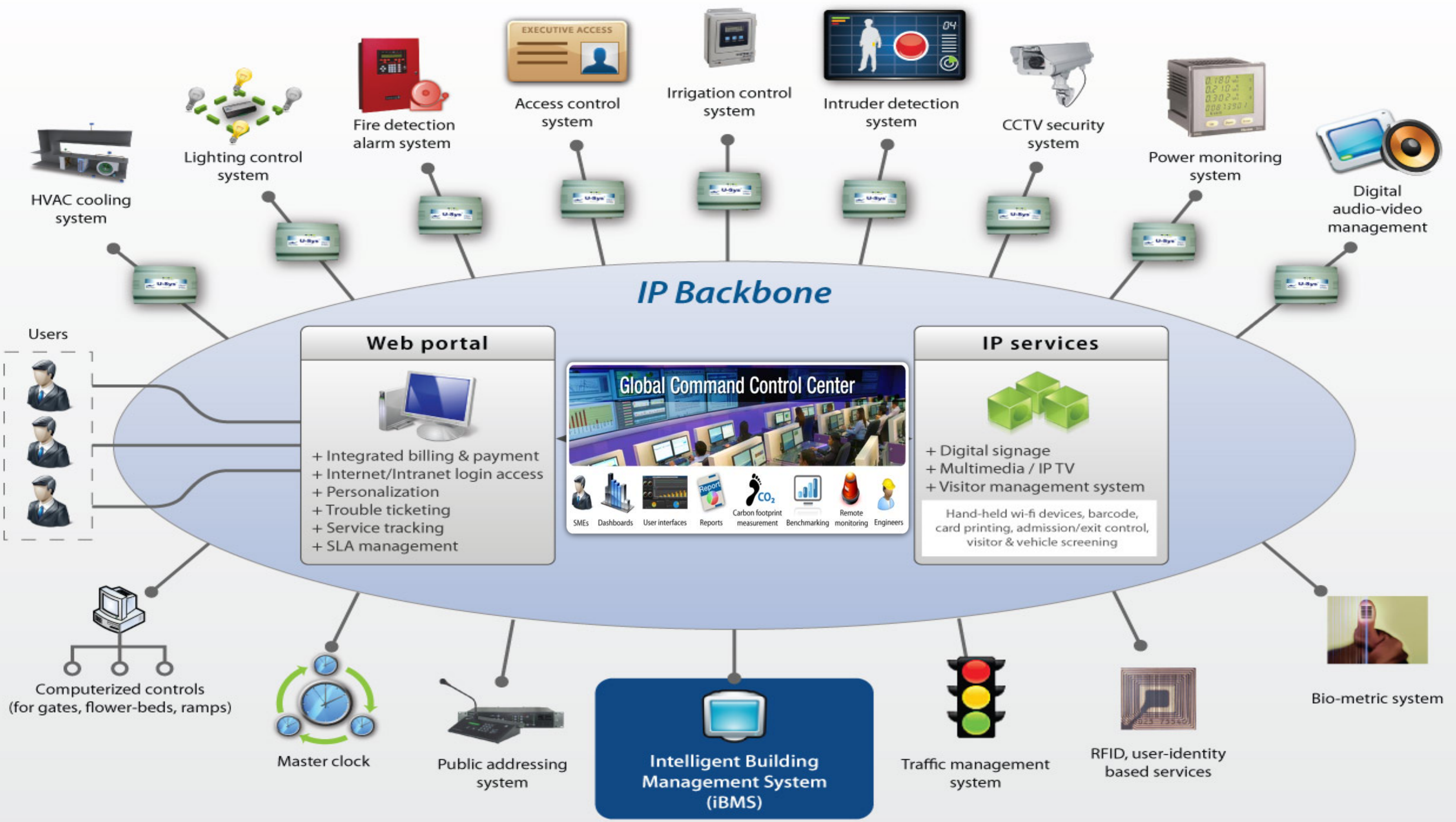
Integration
Presentation
Third Party Application Developers
Service Providers
Facilities Management
End Users
OEM
Carbon Footprint Analysis & GHG Accounting
Asset Performance Management
Measurement & Verification
Continuous Commissioning
Energy Analysis
Fault Detection & Diagnostics
Dashboards & Reports
ERP
ORACLE
SAP
MIS
Maintenance Management
Alarm Management
Galaxy
Enterprise Application
M2M Ecosystem
ICT Network
Database
Security System
Vehicle Tracking System
Vehicle Tracking Device
Cameras
SIM
Temperature Sensor
Pressure Sensor
AHU
Chillers
ATMs
Fire Alarms
KW Meter
KW Meter
Power Meter
Smoke Detector
Humidity Sensor
Flow Meter
Occupancy Sensor
Signage
Boilers
FAHU
UPS
Pumps
Generators
HVTS

Command Control Center for Managed Services

Advanced software to analyze data; *make it actionable*



Big Data – *Your* Data in *action*



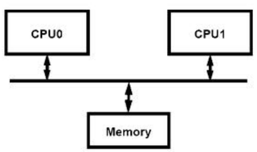
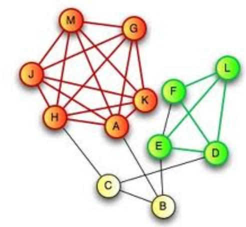
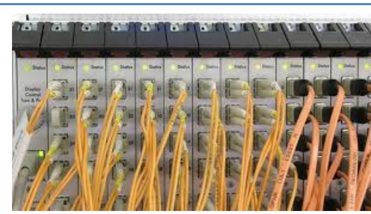
Data G-bots


RETAIL VERTICAL
INDUSTRIAL VERTICAL
LIFE & SAFETY VERTICAL
BUILDINGS
SECURITY & SURVEILLANCE
TRANSPORT VERTICAL
DATA CENTER VERTICAL
HEALTH VERTICAL
HOTELS VERTICAL
AIRPORT VERTICAL
EDUCATIONAL VERTICAL
ENERGY VERTICAL
PACIFIC CONTROLS
Gbots
PACIFIC CONTROLS
Galaxy
Ecosystem
Report
Hybrid Level

Part 2 – Finding Abstractions for Human and Compute Resources for Elastic Systems

Computing Models

S. Dustdar, H. Truong, “Virtualizing Software and Humans for Elastic Processes in Multiple Clouds – a Service Management Perspective”, to appear in *International Journal of Next Generation Computing*, 2012

	Machine-based Computing	Human-based Computing
Processing Unit	  	
Architecture	 SMP   Grid	  
Comm.	 	  

From *Design by Contract*…

Extends conventional software component definitions with

- pre- and post-conditions
- invariants

these specifications are referred to as *contracts*.

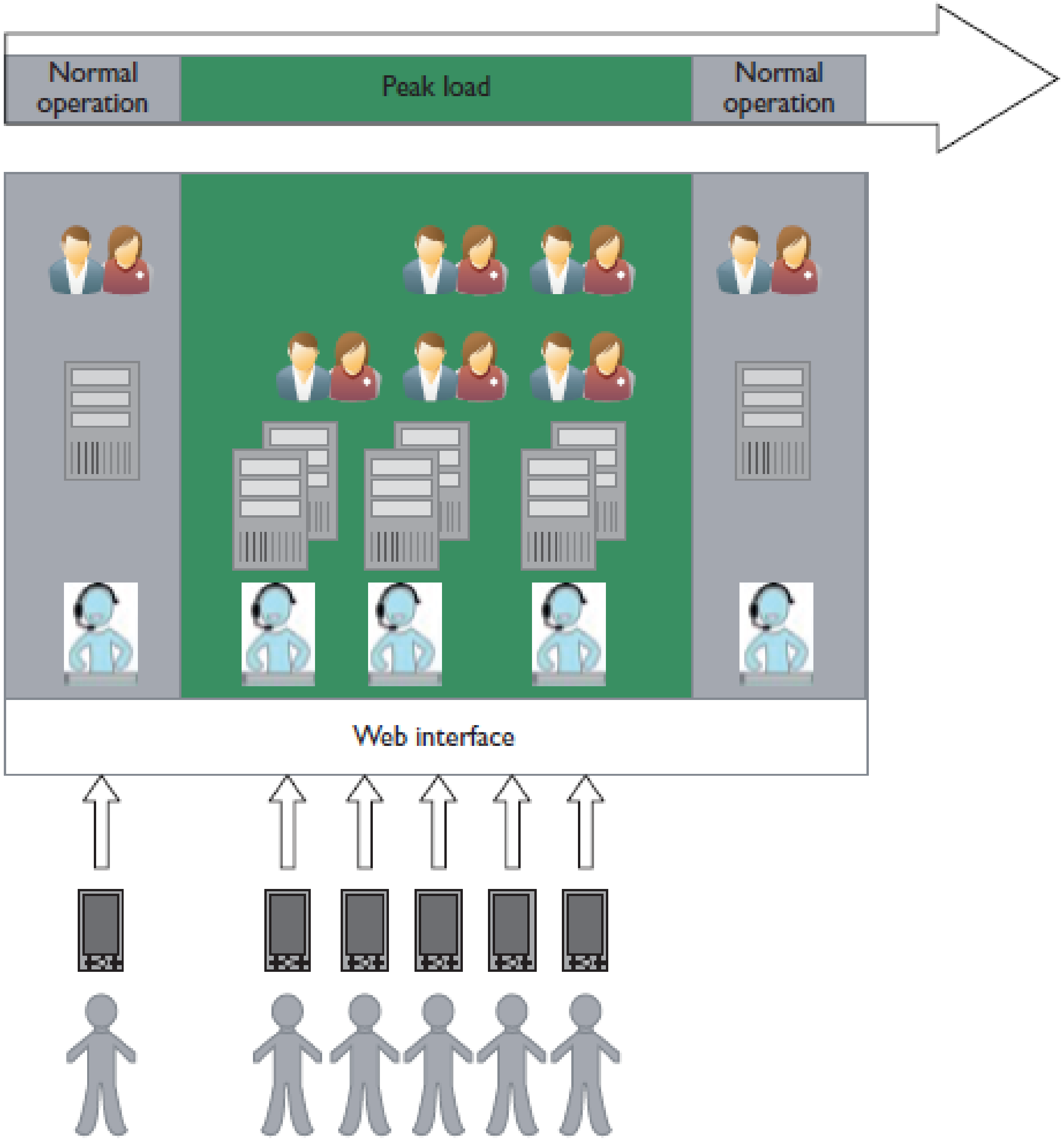
To *Design by Units*

- Extends software service definitions with a resource model to better address

human and compute resource requirements in system design.

- *Units* are abstractions to model diverse resources in the cloud that are required to operate a system and to guarantee nontrivial system requirements, such as **elasticity**.

Elasticity of Resources



Elasticity in computing – a broad view

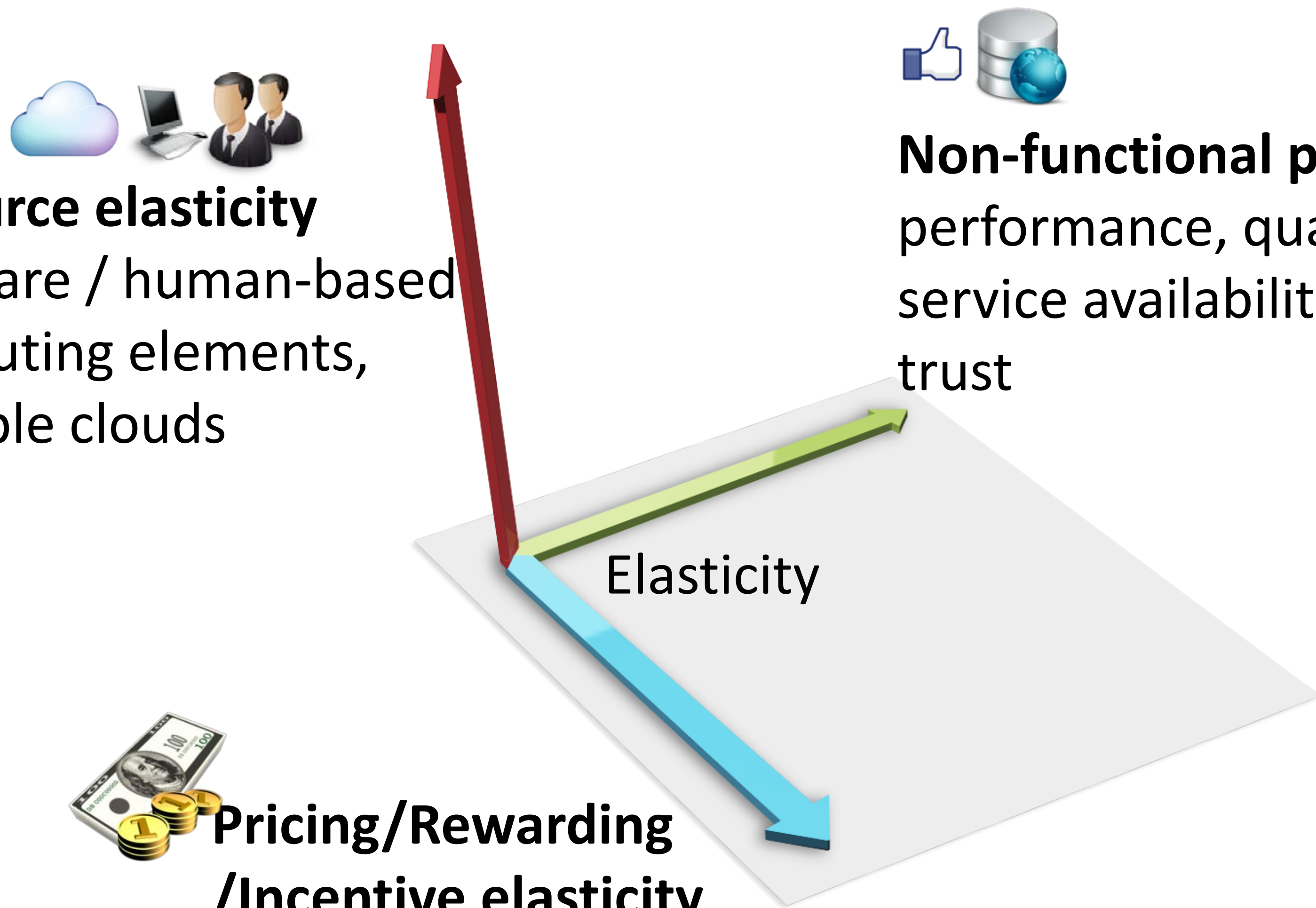
1. Elastic demands from consumers

2. Multiple outputs with different price and quality (output elasticity)

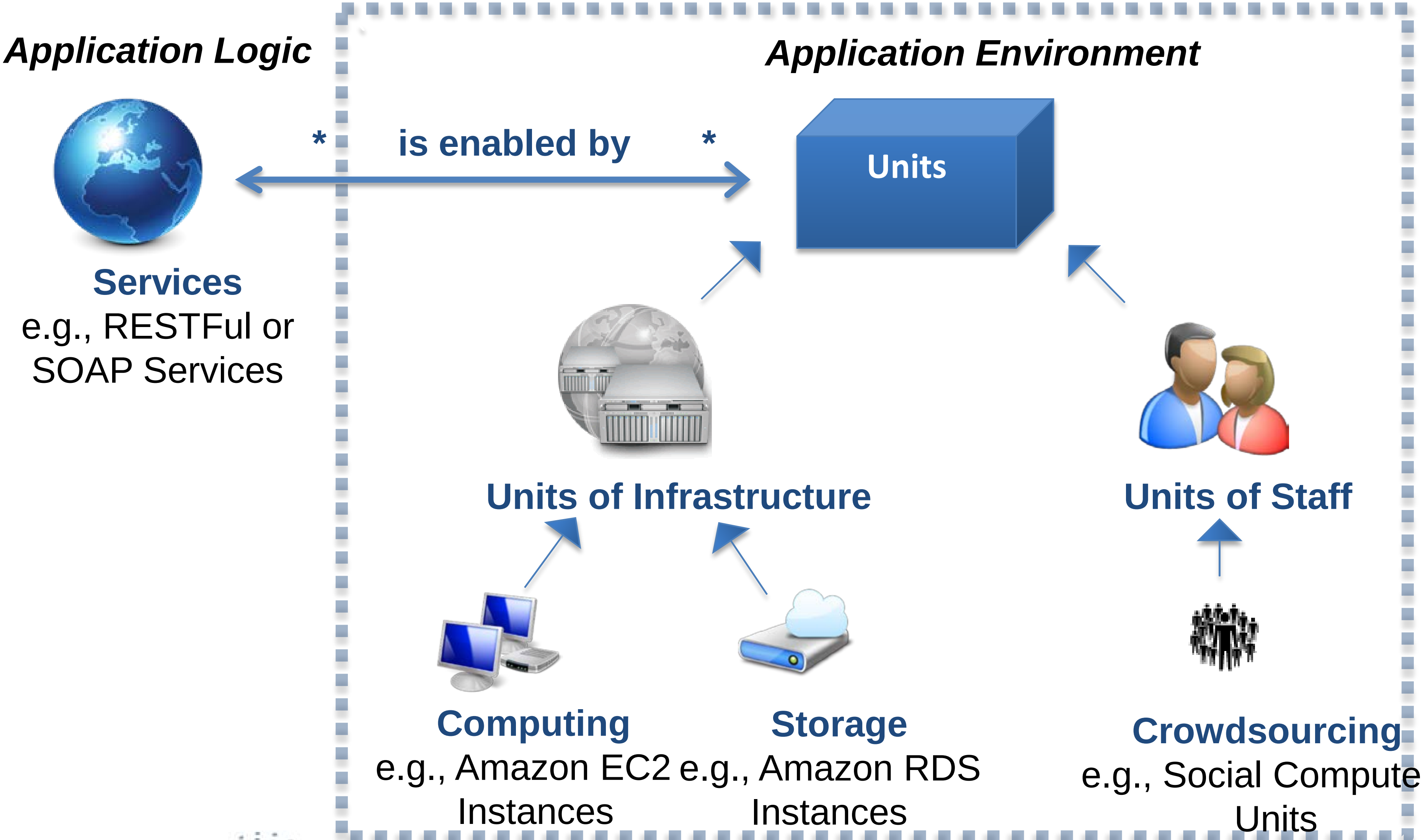
3. Elastic data inputs, e.g., deal with opportunistic data

4. Elastic pricing and quality models associated resources

Elasticity


Resource elasticity
Software / human-based computing elements, multiple clouds
Non-functional parameters
performance, quality of data, service availability, human trust
Elasticity
Pricing/Rewarding /Incentive elasticity

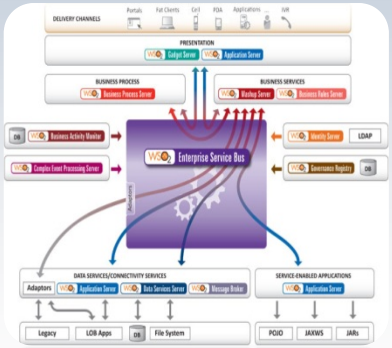
Elastic Units



Unit Types - Examples

Social Resources

- (Teams/Masses of) People
- Using majority votes
- Units as crowds' structuring mechanism



Middleware Platform Resources

- Units as application containers
- Units as user workspaces

Legal Framework Resources

- Units might model compliance policies
- Units might model contracted license keys

Financial resources

- Units to represent funding sources within an organization
- Units to represent funding sources from external sources

Impact

- *Each unit provides a resource to the application (functional)*
- *E.g. impact on availability of the call center, storage units on data management, etc.*

Impact Measurability

- Measure the impact (e.g. on metric scale) for each unit.
- Measure call center availability via average response times

Cost Measurability

- Units consumption costs
- Unit usage costs must be measurable
- Preemptively reason over the benefits and value of unit configurations

Dynamicity

- Acquire and release new units in a timely, on-demand fashion
- Model runtime reactions to changes in the application's environment
- Resource management might be automated or involve human interactions

Unit Specification

Design of Units

- Units provide a virtual representation of resources that can be monitored, reserved, bought, sold, and used
- They have current state
  - cost, utilization, quality, correctness, efficiency, location …
- And they provide operations
  - store in memory unit, compute in computational unit, send in network unit, …

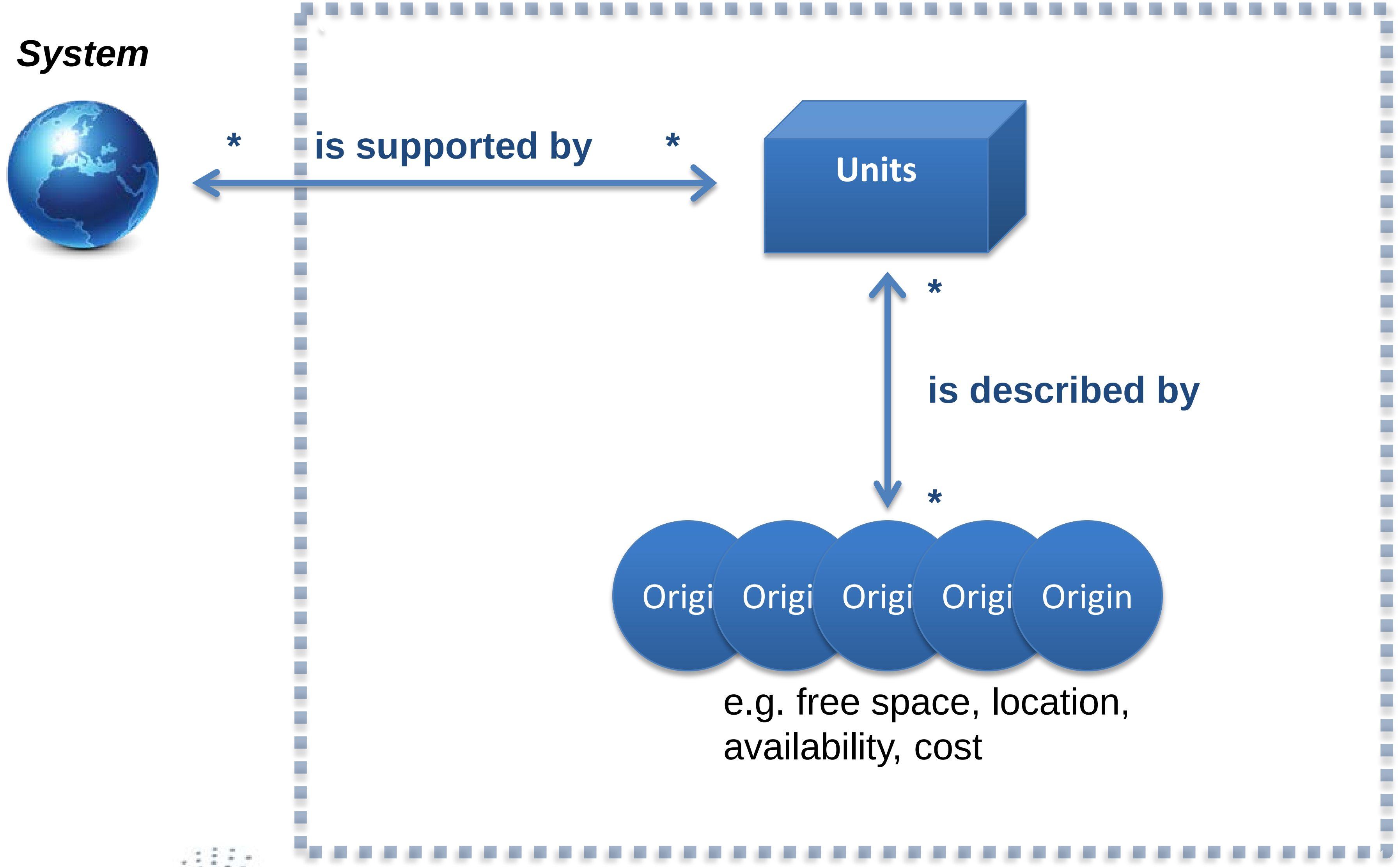
Origins Model (current work)

- Origins provide a programming model for large-scale heterogeneous systems
- An origin
  - represents a single property of a unit
  - which can be composed with other origins to provide higher-level information about the state of a system

Origins Model (current work)

- Thus, we can use origins to
  - build a system
  - that can reason about its current state and adapt to sudden changes
  - by gathering information about its units
- This in turn allows us to guarantee non-trivial requirements in the system
  - e.g. by adapting the system to use more or less units (i.e., provide elasticity)

Units & Origins



Example Code

```
// composition
origin.create("memory/utilized") {
  val memoryUnits: Array[Unit] =
   system.unit("memory")
  val available: Long =
    memoryUnits map {x => x.get("size")} sum
  val used: Long =
    memoryUnits map {x => x.get("use")} sum

   return used / available.toDouble
}

// adaptation
origin.access("memory/utilized" when {_ > .95}) {
   system.unit += market.buy("unit/memory")
}
```

Some papers…

1. **Design by Units,** S. Tai, P. Leitner, S.Dustdar. (2012*). IEEE Internet Computing*, Volume 16, Issue 4 (July/August)

2. **Principles of Elastic Processes,** S. Dustdar, Y. Guo, B. Satzger, H. Truong (2011**),** *IEEE Internet Computing*, Volume 15 (2011), Issue 5; S. 66 - 71.

3. **The Social Compute Unit**, S. Dustdar, K. Bhattacharya (2011). IEEE Internet Computing , Volume 15 (2011), Issue 3; S. 64 - 69.

4. **Trust-based Discovery and Interactions in Mixed Service-Oriented Systems**, Schall D., Skopik F., Dustdar S. IEEE Transactions on Services Computing (TSC), Volume 3, Issue 3, pp. 193-205

5. **Modeling and Mining of Dynamic Trust in Complex Service-oriented Systems**, Skopik F., Schall D., Dustdar S. Information Systems Journal (IS), Volume 35, Issue 7, November 2010, pp. 735-757. Elsevier.

6. **Programming Human and Software-Based Web Services**, Schall D., Dustdar S., Blake M.B. IEEE Computer, vol. 43, no. 7, pp. 82-85, July 2010.

7. **Unifying Human and Software Services in Web-Scale Collaborations** Schall D., Truong H.-L., Dustdar S., IEEE Internet Computing, vol. 12, no. 3, pp. 62-68, May/Jun, 2008.

8. **Runtime Behavior Monitoring and Self-Adaptation in Service-Oriented Systems**, Psaier H., Juszczyk L., Skopik F., Schall D., Dustdar S. 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'10), 27 Sept.-01 Oct. 2010, Budapest, Hungary.

Thanks for your attention

Prof. Dr. Schahram Dustdar

Distributed Systems Group
TU Wien