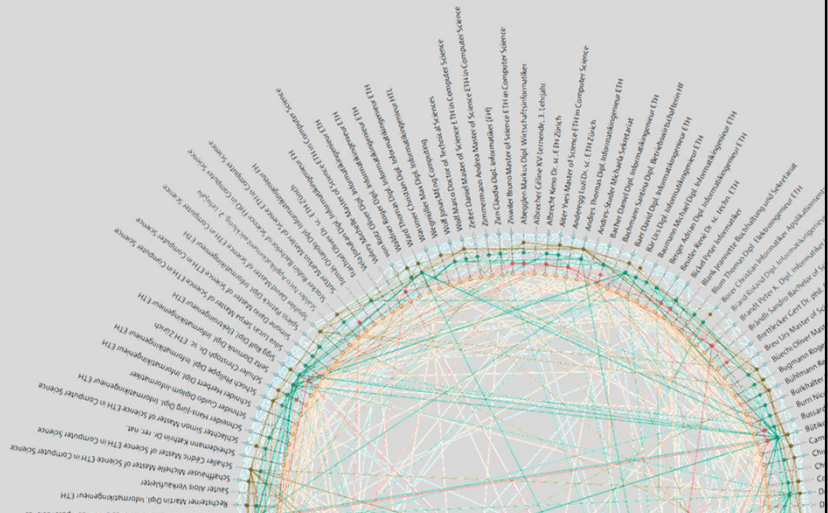




Bereit für das Internet der Dinge

Konzepte für eine vernetzte Zukunft mit embedded Java

Kaspar von Gunten
Ergon Informatik AG



Agenda

Vortrag (40')

- Belimo Energy Valve: Ein produktives Java-basiertes Embedded System
- Shared Logic Plattform
- Komponenten als Basiskonzept
- Herausforderungen und Erfahrungen
- Ausblick

Live-Demo (10')

- Intelligente Klimatisierung

Q & A (10')

Belimo Energy Valve



5

ergon

Belimo Energy Valve

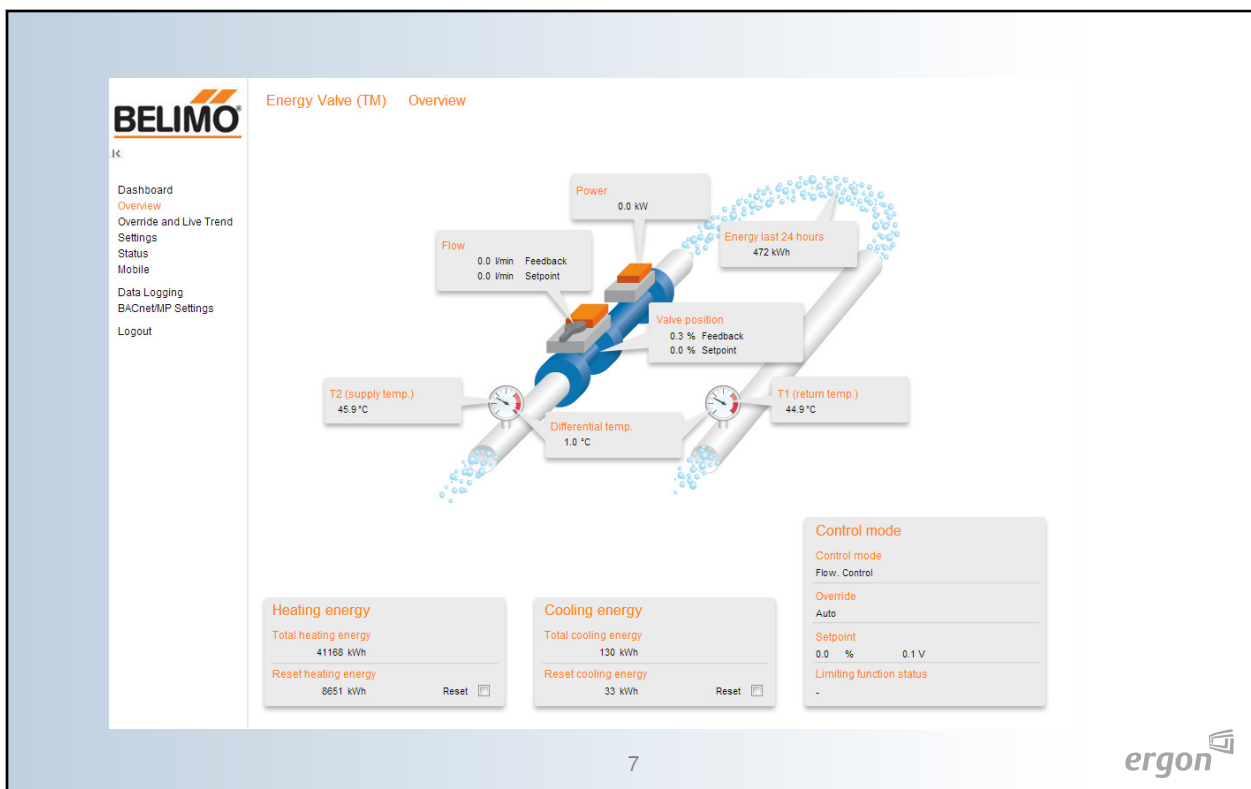
- Das Belimo Energy Valve™ ist ein intelligenter Ventil-Stellantrieb, der weltweit für effizientes Heizen und Kühlen von Gebäuden eingesetzt wird.
- Das Ventil **optimert** den Wasserfluss **lokal** und **unabhängig** von der Leitstelle.
- Kann durch ein **Web-Interface** parametrisiert und kontrolliert werden.



Belimo (<http://belimo.ch>) Belimo entwickelt, produziert und vertreibt seit 1975 elektrische Antriebe für Luftklappen und Armaturen für die Heizungs-, Lüftungs- und Klimatechnik (HLK) und ist in diesem spezifischen Markt international an der Spitze. Belimo hat ihren Hauptsitz in Hinwil und hat weltweit etwa 1200 Mitarbeiter.

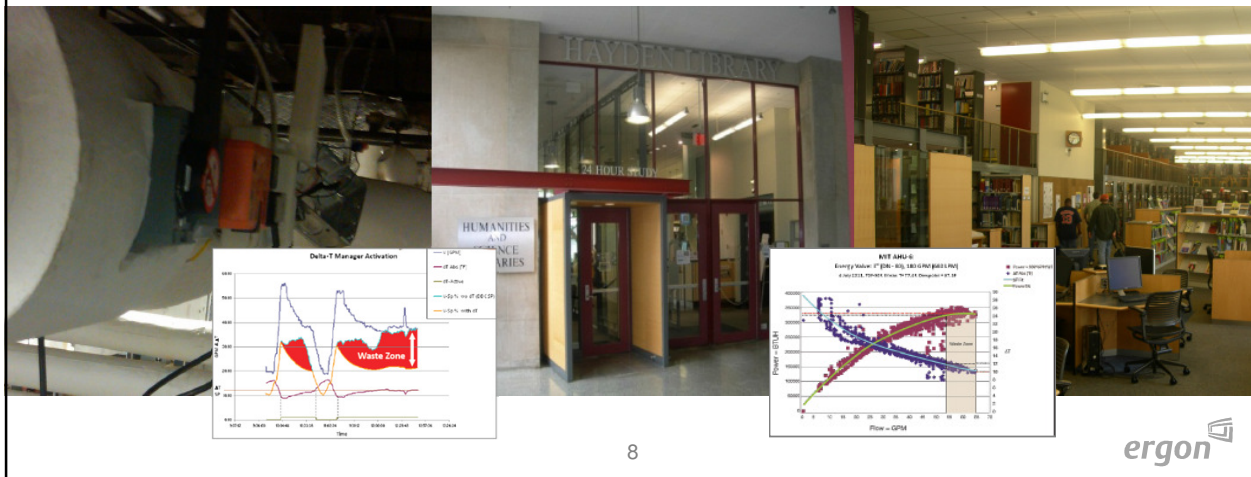
6

ergon



Charles Hayden Bibliothek (MIT)

Dank Retrofitting mit Belimo Energy Valve spart das MIT rund 1.5 Millionen Dollar pro Jahr.



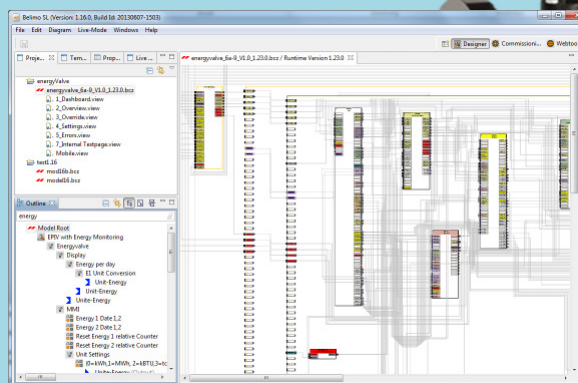
« What impressed us was having such intelligence right on the valve actuator. You can characterize a coil's performance with just a couple of pieces of data. »

Peter Cooper
Manager of Sustainable Engineering and Utility Planning at MIT

9

ergon

Shared Logic

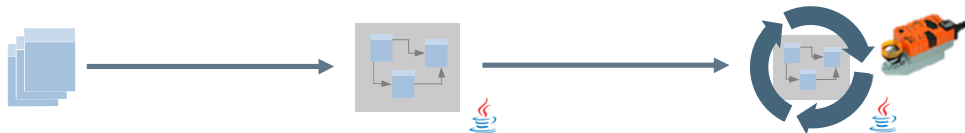


10

ergon

Shared Logic

Die Steuerungs-Software des Energy Valve wurde mit einer generischen, Java-basierten **Entwicklungsumgebung** für embedded devices erstellt.



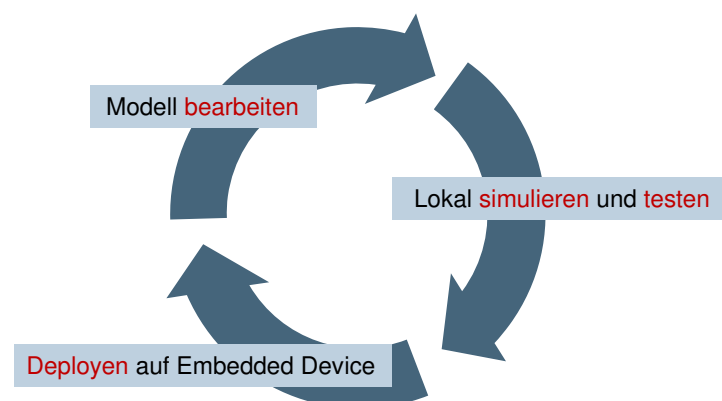
- Erweiterbare **Komponentenbibliothek**
- Grafischer Editor für **visuelle Modellierung** der Steuerungssoftware
- **Ausführungslogik (Engine)** auf den eingebetteten HW-Knoten

11

ergon 

Modellentwicklung

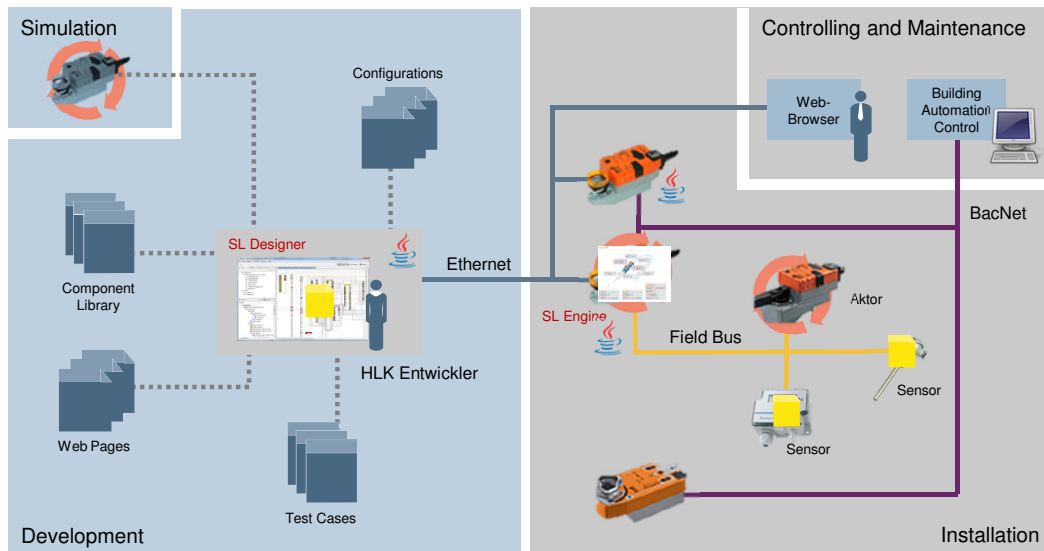
Shared Logic ermöglicht **Rapid Prototyping** im Embedded System-Bereich.



12

ergon 

Entwicklungs- und Laufzeitumgebung



13

ergon

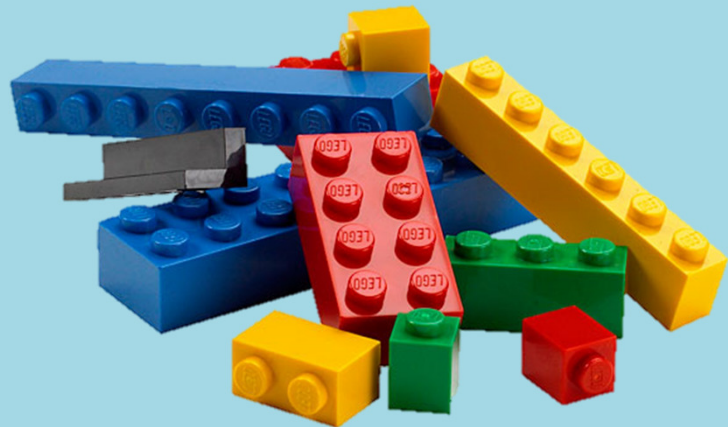
Eigenschaften

- Steuerungslogik (= Modell) in XML
- Ausführungslogik (= Engine) in Java
- Kommunikation mit Sensoren und Aktoren (MP)
- Standardisierte Kommunikationsprotokolle (BacNet)
- Kommunikation zwischen Knoten über IP
- Web-Interface (Jetty)
- Läuft auf jedem Linux-basierten System mit Oracle J2SE Embedded JVM
- Grafischer Editor (Eclipse RCP)
- Testen und Simulieren
- Laufzeit-Inspektion und Deployment

14

ergon

Komponenten



15

ergon

Modulare Entwicklung für Eingebettete Systeme

Komponenten sind ein zentrales Konzept in der klassischen SW-Entwicklung: Modularität vereinfacht Erweiterung, Wiederverwendung und Unterhalt.

Plattform

- Einsatz von Modularität in mehreren Dimensionen

Anwendungsentwicklung für eingebettete Systeme

- IDE unterstützt und fördert die Entwicklung von modularen Applikationen

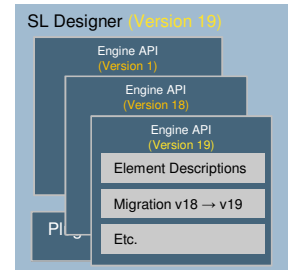


ergon

Plattform-Komponenten

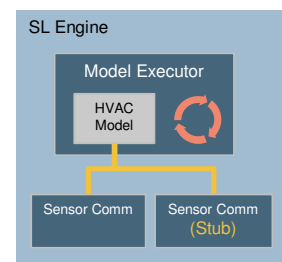
SL Designer

- Eclipse **Plugin**-Architektur
- API-Packages unterstützen alte **SL Engine Versionen**



Embedded SL Engine

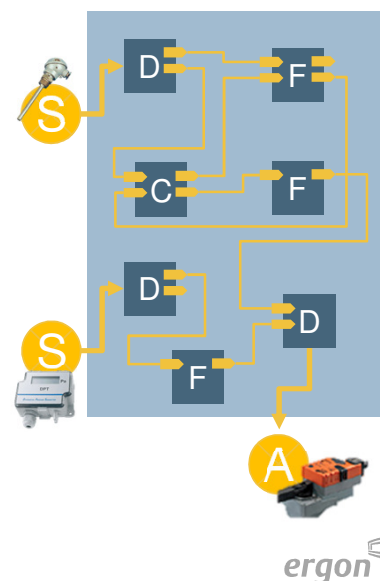
- Besteht aus **Bundles** und **Services** (Felix OSGi)
- **Alternative Implementationen** für Test und Simulation



17

Applikationsentwicklung für Einzelknoten

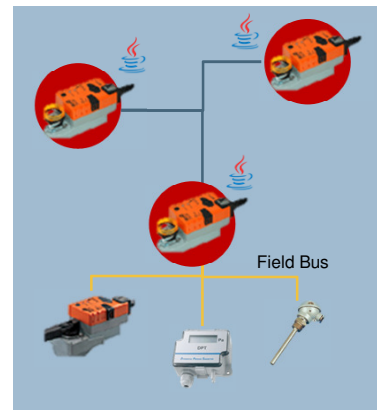
- **Sensoren und Aktoren** können durch neue **Treiber**-Komponenten eingebunden werden
- **Transformationen und Berechnungen** werden durch neue **Funktions**-Komponenten hinzugefügt
- Teile eines existierenden Modells können zu wiederverwendbaren **Compound**-Komponenten zusammengefasst werden
- Compounds können in **Bibliotheken** gesammelt werden
- Für Compounds können **Unit Tests** erstellt werden



18

Applikationsentwicklung für Knotenverbände

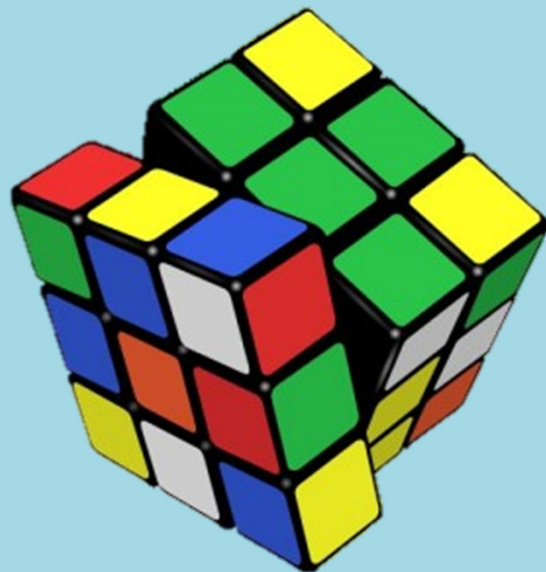
- Kollaborative Multi-Knoten-Logik durch Kommunikation über Ethernet
- Einzelne Knoten agieren als **Manager** für Sensoren und Aktoren in lokalem Feld



19

ergon 

Herausforderungen



20

ergon 

Herausforderung 1: Robustheit und Stabilität

24/7 Laufzeit, angenommene TTL 20 Jahre

Problem: Erfordert extensive Tests für Langzeitverhalten und -Nebeneffekte

- Einrichten von **Regressionstests** und **Langzeittests** auf End-HW
- Einsatz von **SWE best-practices** um hohe Qualität zu garantieren
- Verwendung von **stabilen** OS und JVM

Problem: Geräte müssen nach **Fehlfunktion** oder **Stromausfall** selbstständig in stabilen Zustand zurückfinden

- **Watchdog Services** überwachen Systemzustand und lösen Reset aus

21



Herausforderung 2: Unterhalt

Problem: Nach Verbauung sind Geräte oft schwer (oder gar nicht) zugänglich

- Einrichten von **Remote Monitoring** und **Software Patching**

Problem: Eingebettete SW erfordert Langzeitsupport für jeden Release.

- Alle je erstellten **SW-Artefakte müssen jederzeit zugänglich** sein
- **Rückwärtskompatibilität** der IDE für Modelle (Featurelock)
- Nur **etablierte Technologien** verwenden

22



Herausforderung 3: Limitierte Hardware

64 MB RAM, 64 MB Flash Memory, 200 MHz CPU

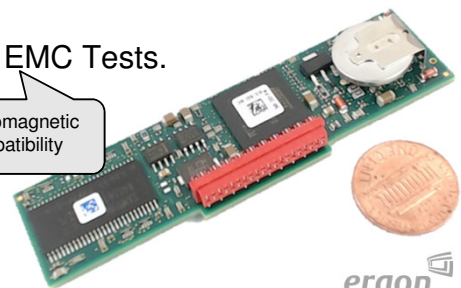
Problem: Abwägung zwischen Speicherverbrauch und Antwortzeit.

- Speicherverbrauch beobachten
- Saubere Konfiguration von OS, FS und JVM

Problem: Wechsel von Hardware erfordert teure EMC Tests.

- Defensive HW Upgrades

Electromagnetic
compatibility



ergon

23

Erkenntnis 1: Kosten vs. Flexibilität

Einfache "dumme" HW wird nie komplett verschwinden, dafür ist sie zu billig. Aber...

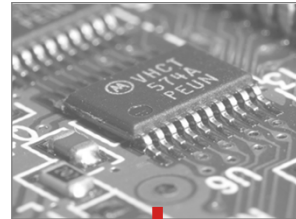


ergon

24

Erkenntnis 2: Erhöhte Software-Komplexität

- Die zunehmende Intelligenz von eingebetteten Systemen führt zu **Wechsel von Hardware zu Software Engineering** in der Entwicklung.
- **Entwickler von Embedded Devices** müssen zwingend in SWE Praktiken geschult werden.
- Ein **adäquater SW Entwicklungsprozess** ist eine zentrale Anforderung.
- Erhöhte Software-Komplexität verlangt nach **modularem Design**.



↓

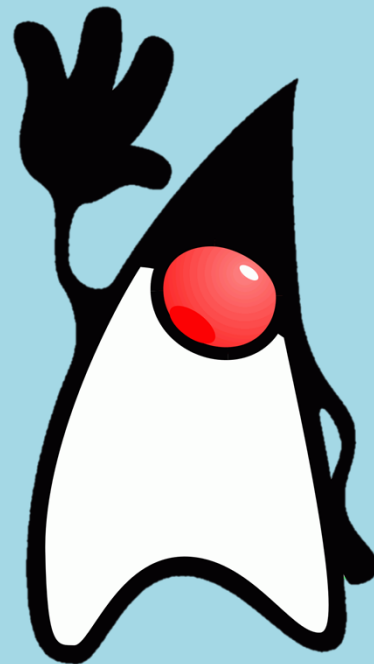
```
final Integer busAddress=
final MpDeviceIdentifier m
this.accessor.unsetBusAddr

int cnt= this.accessor.get
do {
    final MpDeviceIdentifi
    // no device anymore a
    if (id == null || id.e
```

25

ergon

Ausblick



26

ergon

Das Internet der Dinge ist nah!

- Immer mehr Kleinstgeräte werden Teil eines Netzwerks
- SW-Intelligenz rückt zusehends näher zu Sensoren und Aktoren
- Als Konsequenz davon wird unsere Umwelt immer smarter

Unsere Erfahrung zeigt klar, dass **Java ein solides Fundament** für Entwicklungen und Produkte in diesem Bereich bietet.

Oracle hat die **offizielle Strategie**, Java als Basistechnologie für das Internet of Things zu etablieren.

27



« ... this whole enterprise stuff was an accident on the way to the internet of things. »

James Gosling
Java-Daddy, an der Java One 2013

28



Java is your Perfect Companion

Während der letzten 7 Jahre haben wir von vielen Java-Eigenschaften profitiert:

- **Bibliotheken**, die auf sicheren und bewährten Technologien basieren
- **Modularisierung** und **Erweiterbarkeit** durch OSGi
- Umfangreiche Unterstützung für **Testing** und **Debugging**
- Plattformunabhängigkeit
- Mächtige und ausgereifte **Application Frameworks**
- Auswahl an verschiedenen **embedded JVMs**

29

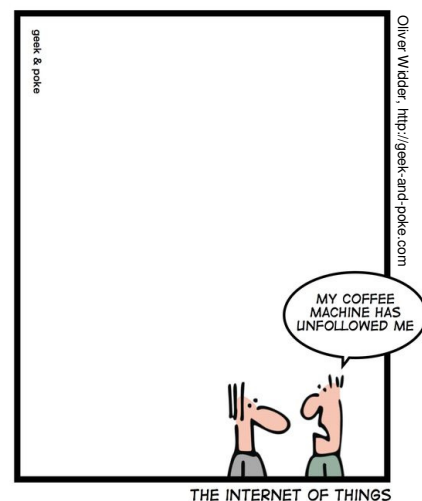


Fazit

Mit

- **Java** als Implementationsplattform
- einer **modularen Architektur** und
- einem **zeitgemässen SW-basierten** Entwicklungsprozess für eingebettete Applikationen

darf das Internet der Dinge mit Zuversicht erwartet werden.



30



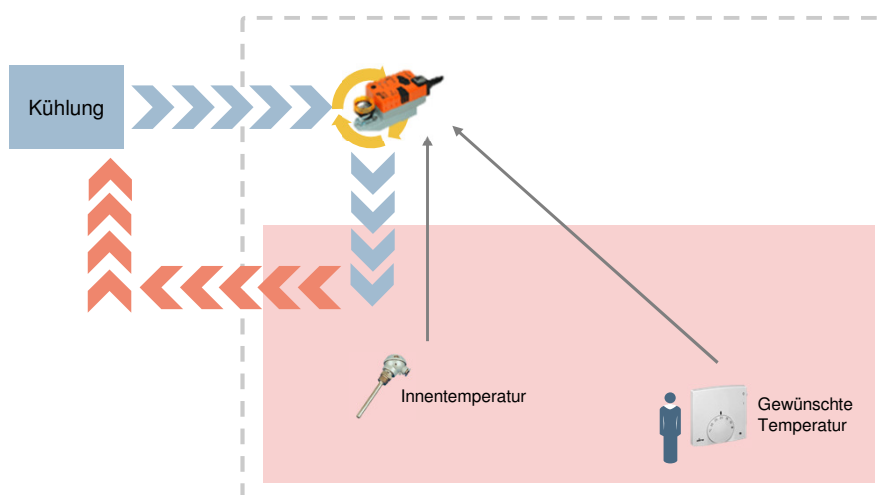
Demo



31

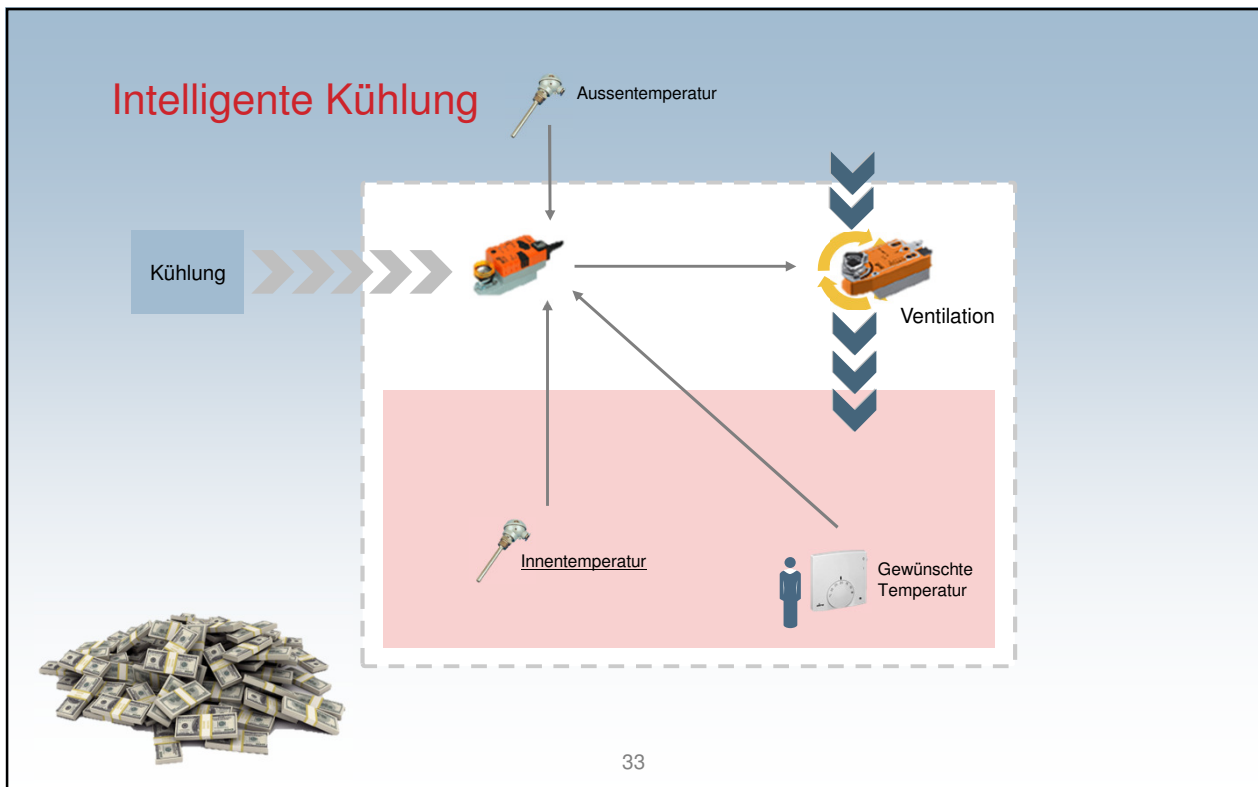
ergon

Raumklimatisierung



32

ergon





Vielen Dank für's Zuhören.

kaspar.vongunten@ergon.ch

www.ergon.ch

twitter.com/ErgonAG

© Copyright 2013, Ergon Informatik AG, Zürich