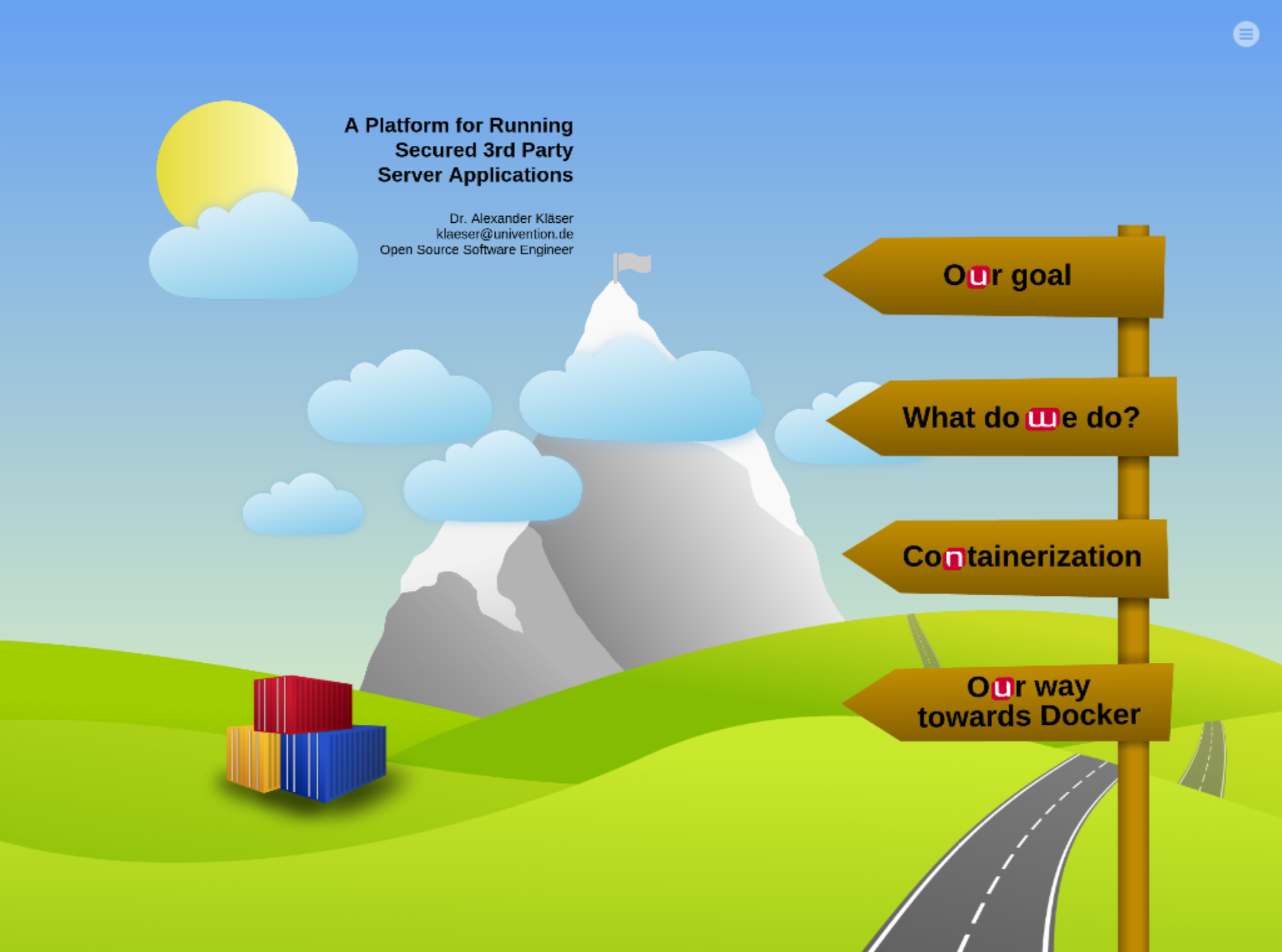
A Platform for Running Secured 3rd Party Server Applications

Dr. Alexander Kläser
klaeser@univention.de
Open Source Software Engineer

A Platform for Running Secured 3rd Party Server Applications

Dr. Alexander Kläser
klaeser@univention.de
Open Source Software Engineer



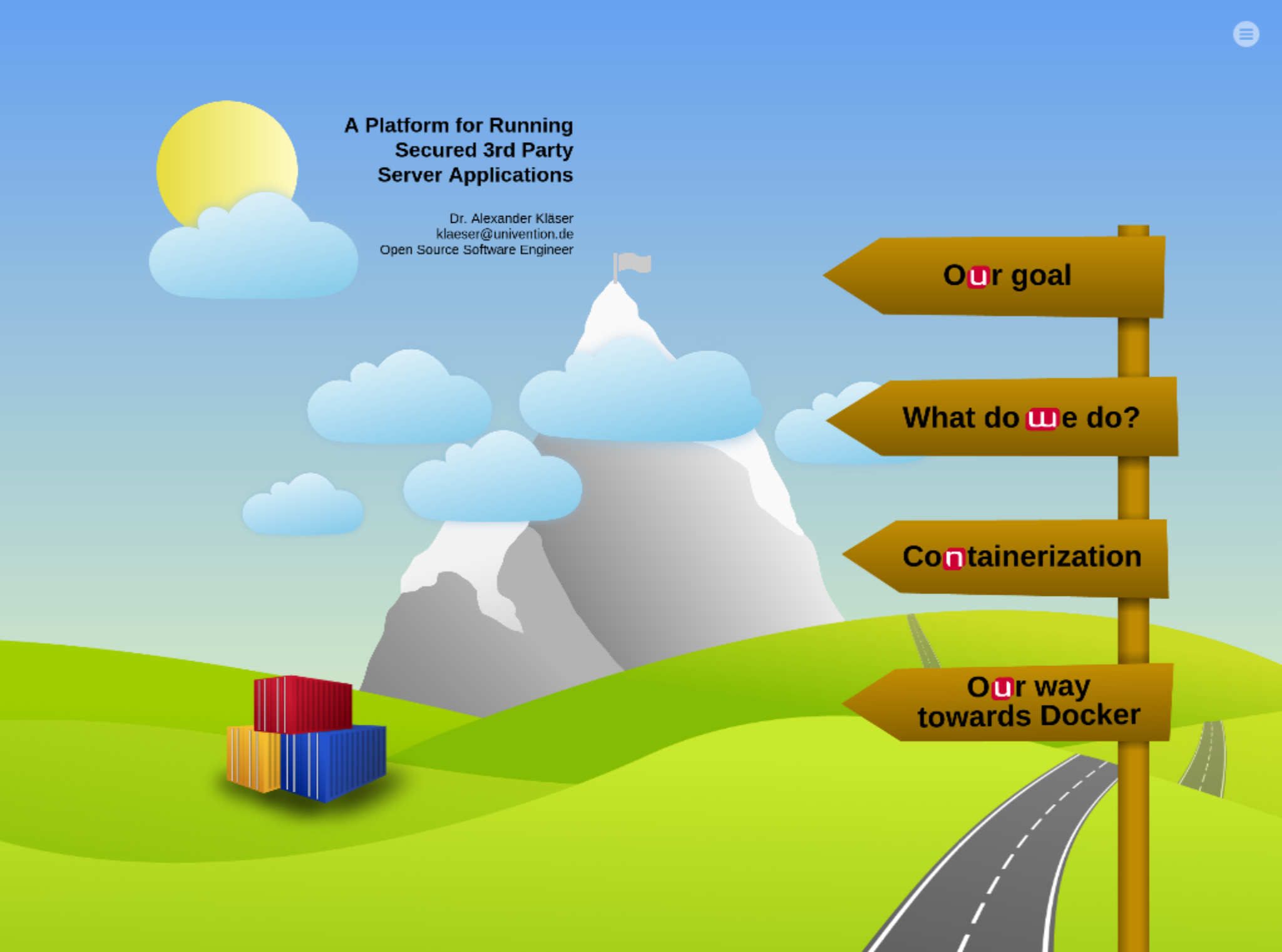

Our goal

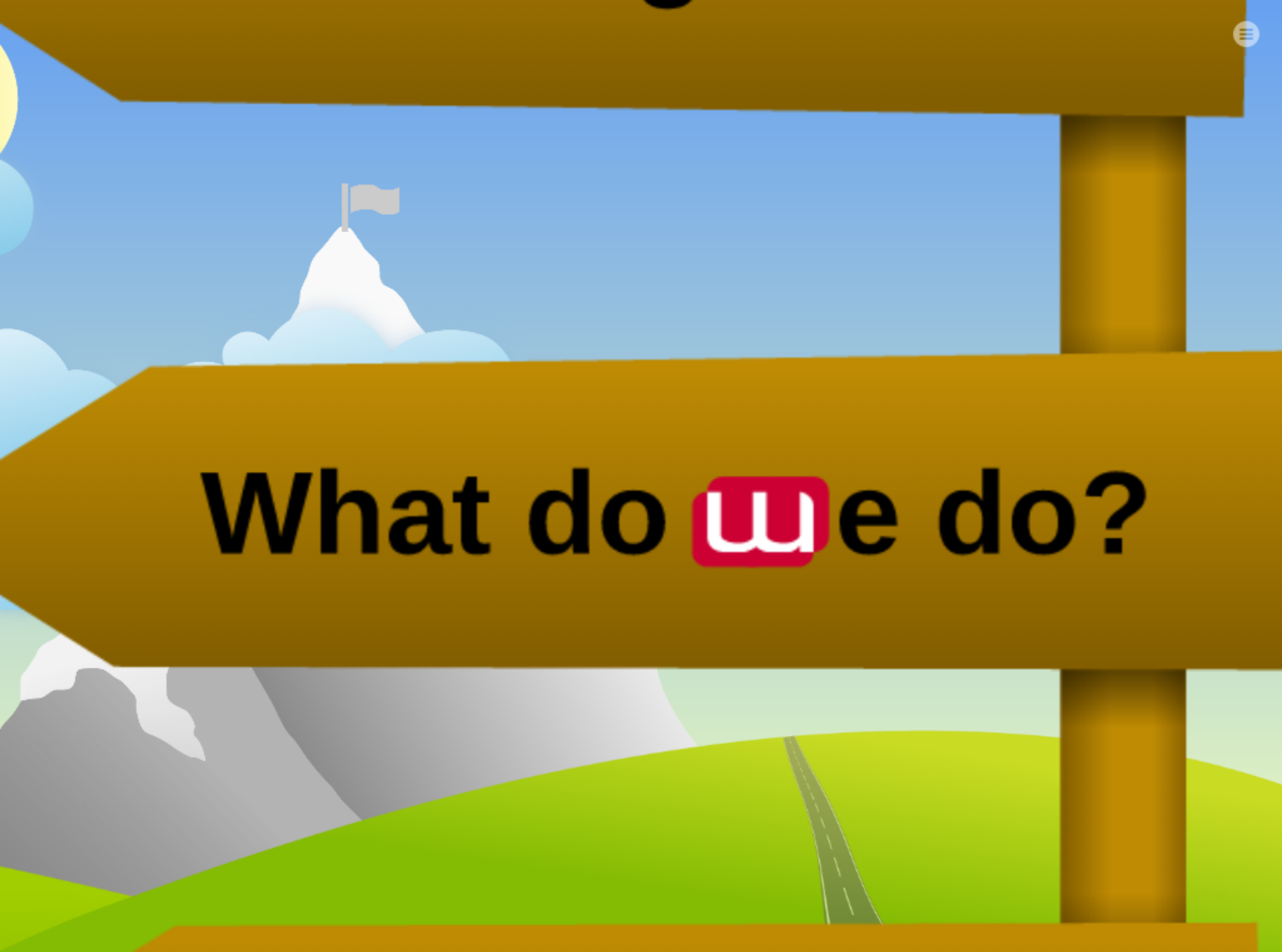
What is our goal?

- Give organizations control over their data and processes
  - Freedom of choice
  - Allow to manage IT in an easy and flexible manner
  - Cost effective and enterprise-ready solutions
  - Leverage innovation potential
- Open Source is the **key element** to achieve this

A Platform for Running Secured 3rd Party Server Applications

Dr. Alexander Kläser
klaeser@univention.de
Open Source Software Engineer



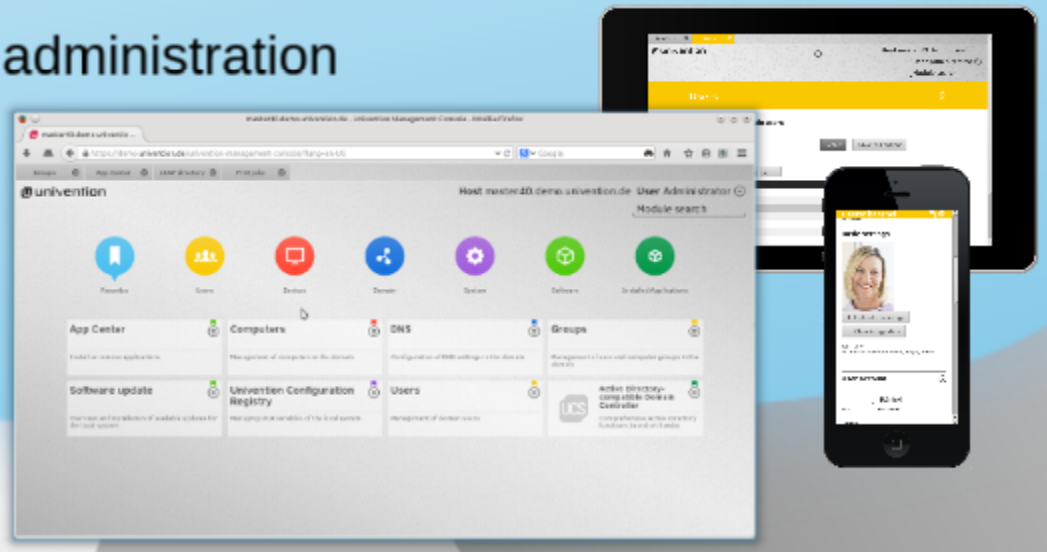

What do we do?

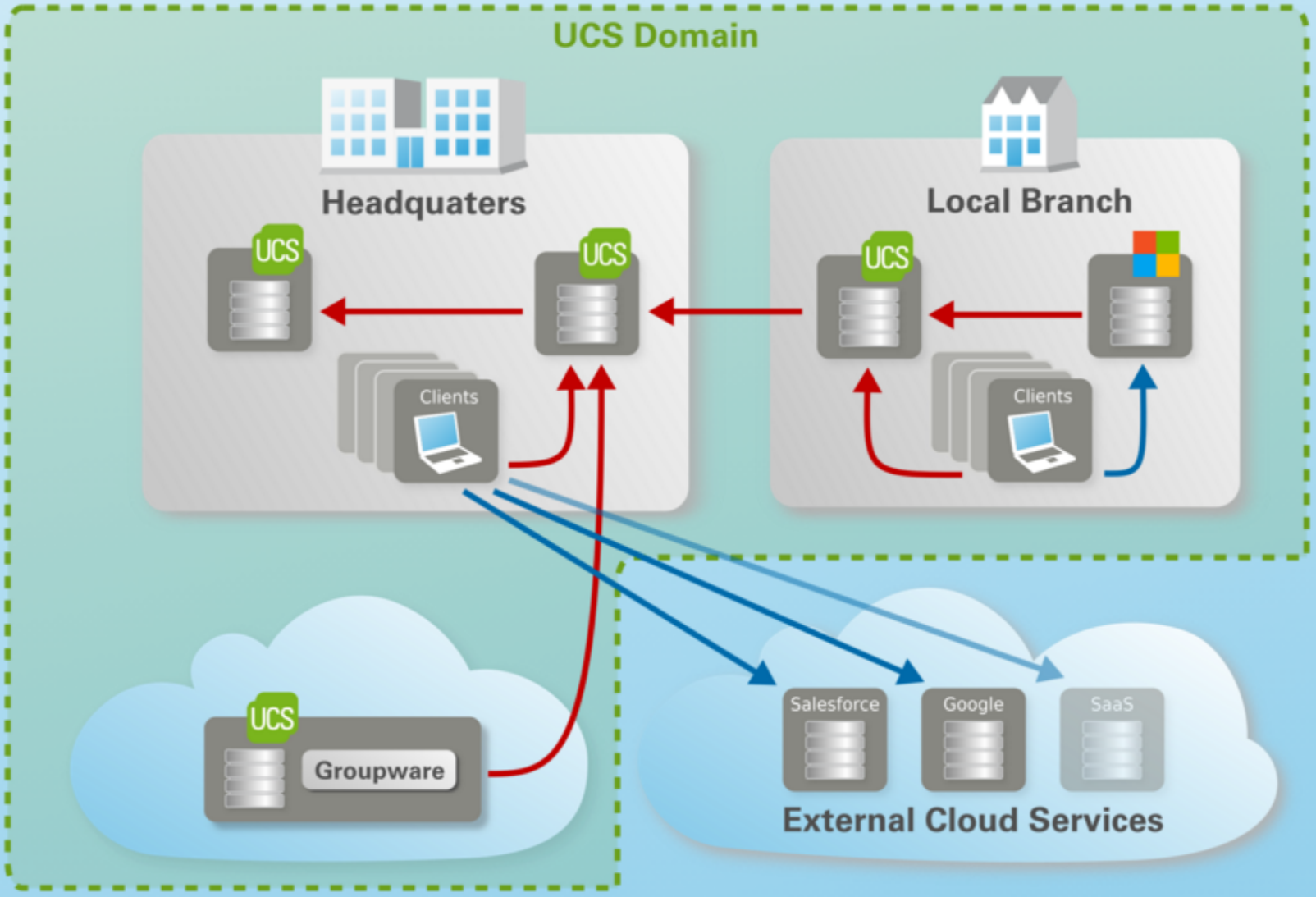
What do we do?

- Main product: UCS Univention Corporate Server
- Identity and infrastructure management
- Platform for 3rd party applications
- 100% Open Source Software
- Cost-free edition available
- Based on Debian Linux

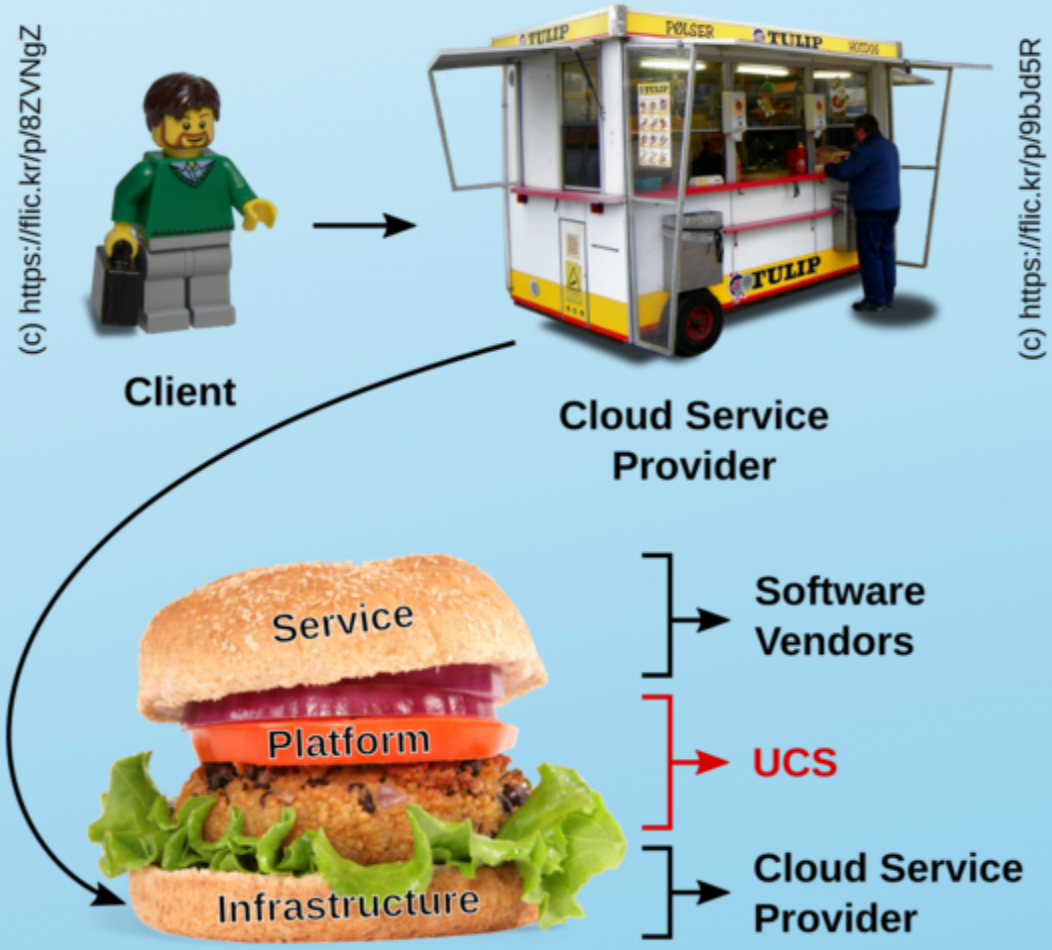
Common features of UCS

- Holistic management of IT infrastructure
- UCS on-premises & in the cloud & hybrid
- Support for Windows, Macintosh, Linux clients
- Single point of administration
- Web interface



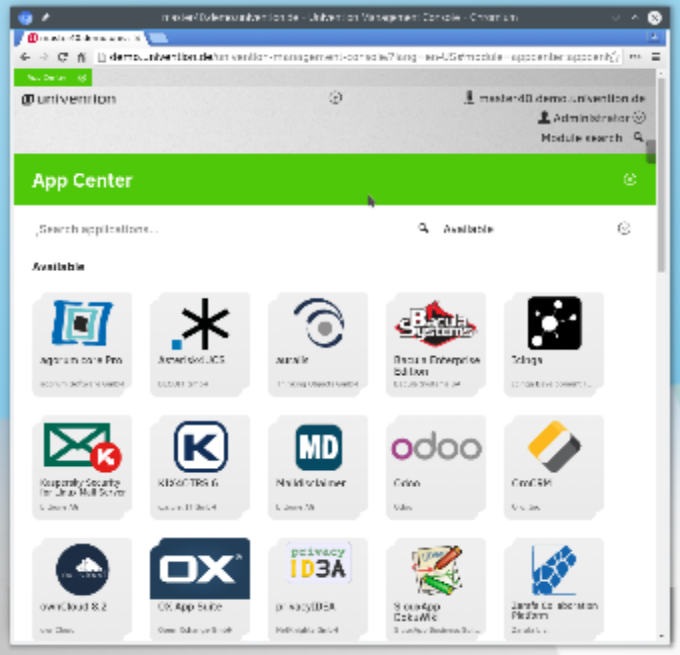

UCS Domain
Headquaters
Local Branch
UCS
UCS
UCS
UCS
Clients
Clients
UCS
Groupware
Salesforce
Google
SaaS
External Cloud Services

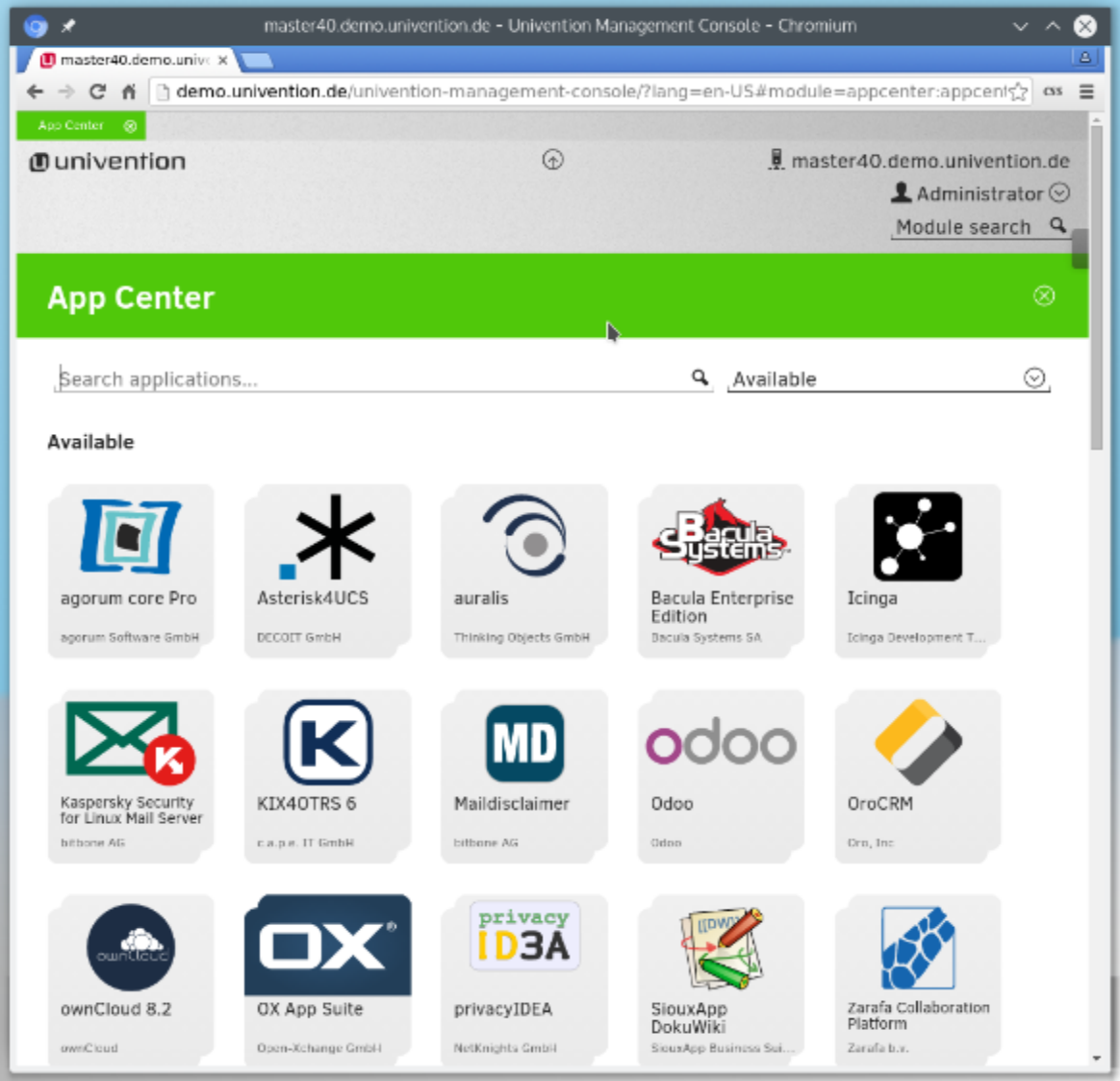
UCS as platform for 3rd party software



Univention App Center

- ... think of UCS as *Android* for servers
- Management via *App Center* & simple installation
- Integration into existing (hybrid) IT infrastructure
- Integration into UCS web interface
- Ecosystem of different solutions



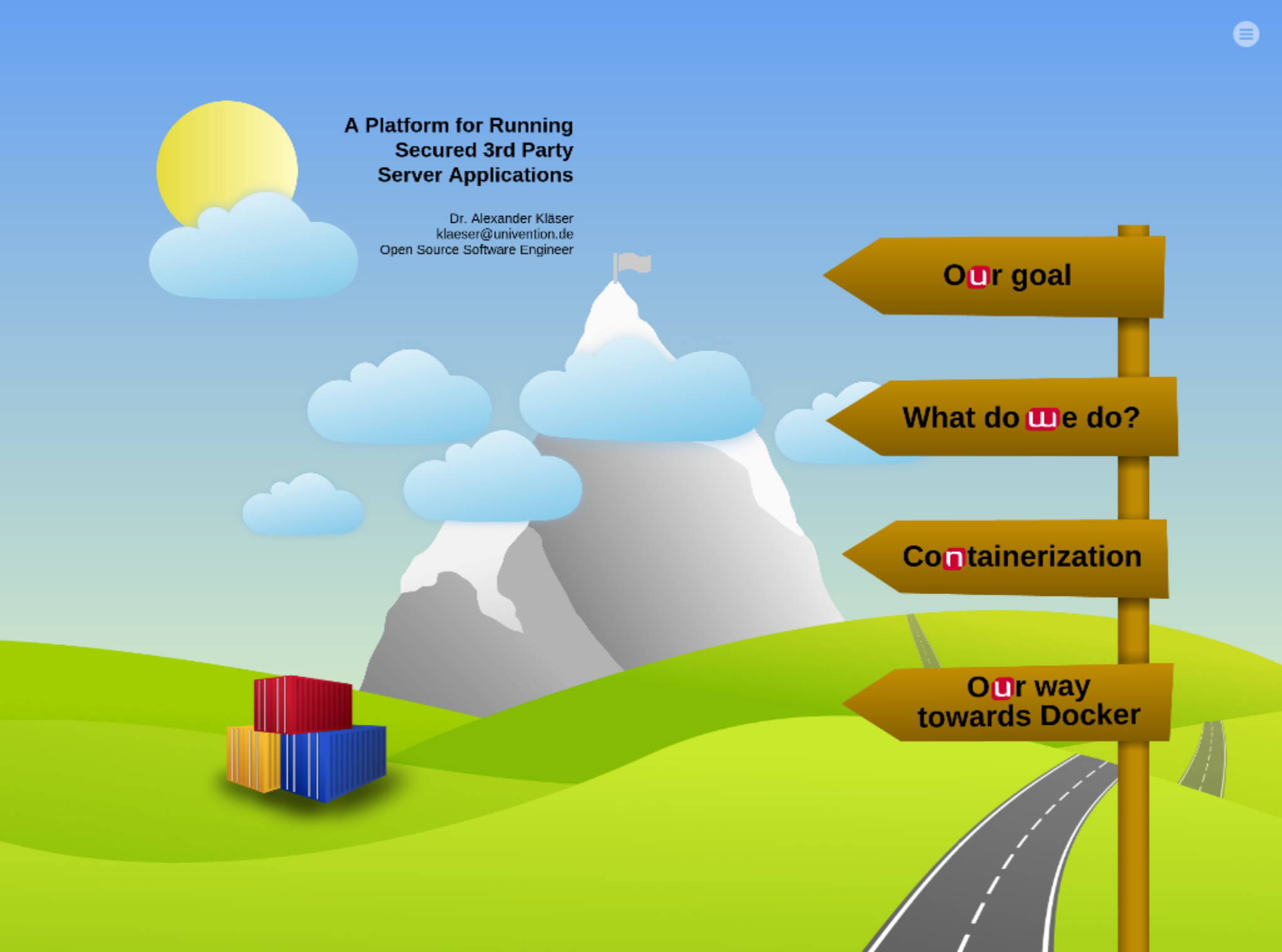

master40.demo.univention.de - Univention Management Console - Chromium
master40.demo.univ
demo.univention.de/univention-management-console/?lang=en-US#module=appcenter:appcent
App Center
univention
master40.demo.univention.de
Administrator
Module search
App Center
Search applications...
Available
Available
agorum core Pro
agorum Software GmbH
Asterisk4UCS
DECOIT GmbH
auralis
Thinking Objects GmbH
Bacula Enterprise Edition
Bacula Systems SA
Icinga
Icinga Development T...
Kaspersky Security for Linux Mail Server
bitbone AG
KIX4OTRS 6
c.a.p.e. IT GmbH
Maildisclaimer
bitbone AG
Odoo
Odoo
OroCRM
Oro, Inc
ownCloud 8.2
ownCloud
OX App Suite
Open-Xchange GmbH
privacyIDEA
NetKnights GmbH
SiouxApp DokuWiki
SiouxApp Business Sui...
Zarafa Collaboration Platform
Zarafa b.v.

Challenges with 3rd party apps

- Continuously growing number of UCS systems
- Continuously growing number of apps (currently ca. 80)
- App needs to be supplied as Debian software package
  - App has full access to UCS system
  - Possible conflicts w.r.t. software dependencies + ports
  - App may intefere with system libraries
  - More apps → increasing complexity
- **Solution:** Containerization (via Docker)

A Platform for Running Secured 3rd Party Server Applications

Dr. Alexander Kläser
klaeser@univention.de
Open Source Software Engineer




Containerization

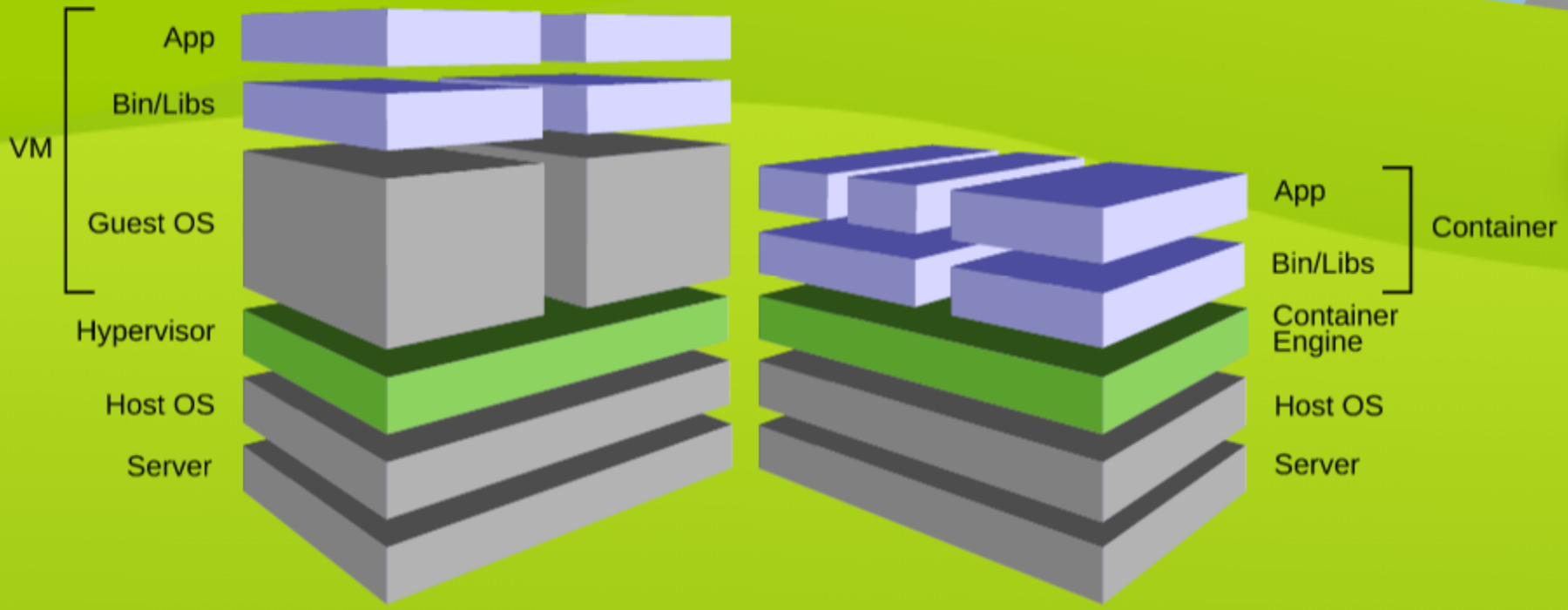
Analogy to transport system

- Decoupling of transport & content
- Standardized containers

Software container

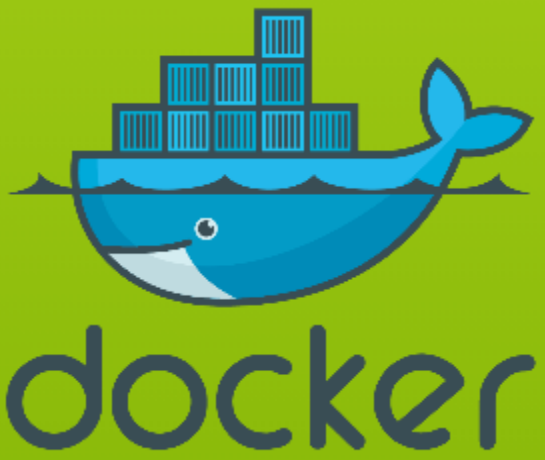
- Standardization of software environment
- Correct execution can be assured
- Software developer is in charge of the inside
- Operator of infrastructure is in charge of the handling
- Linux kernel allows clean sandboxing:
  Control groups + namespaces + capabilities = containers
- Low overhead, container process runs natively in host kernel

VM vs. Container



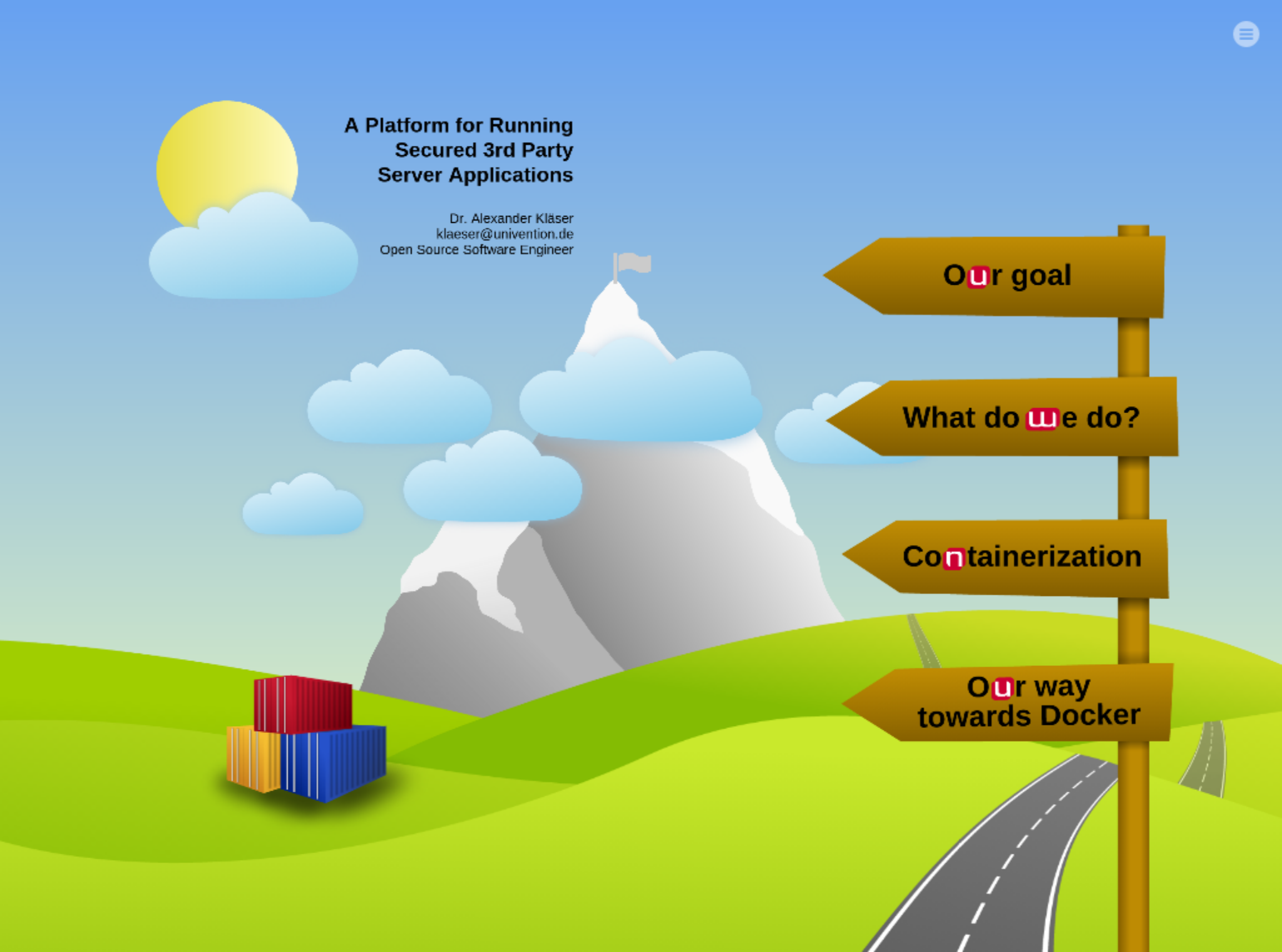
Docker

- Offers tools for efficient usage of Linux kernel container technology (which already exists since 2.6.29)
- Abstracts many details (handling of network, namespaces, cgroups, mounting etc.)
- Docker container starts as a single command
- Container is not booted (*/sbin/init* needs to be called manually)
- Software dependencies are stored on separate RO images
- Containers can share images
- Only first layer is writable (copy-on-write)

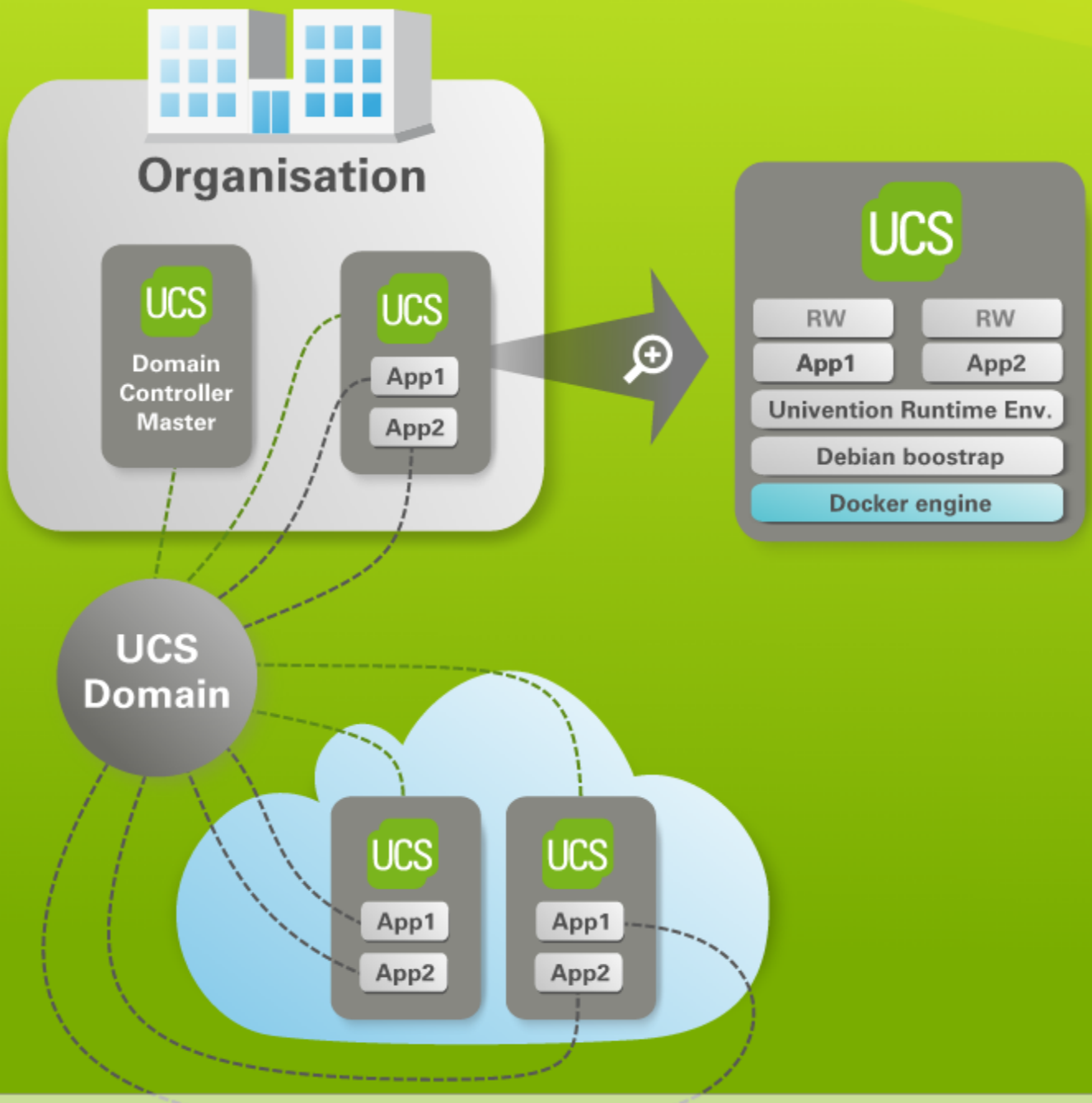


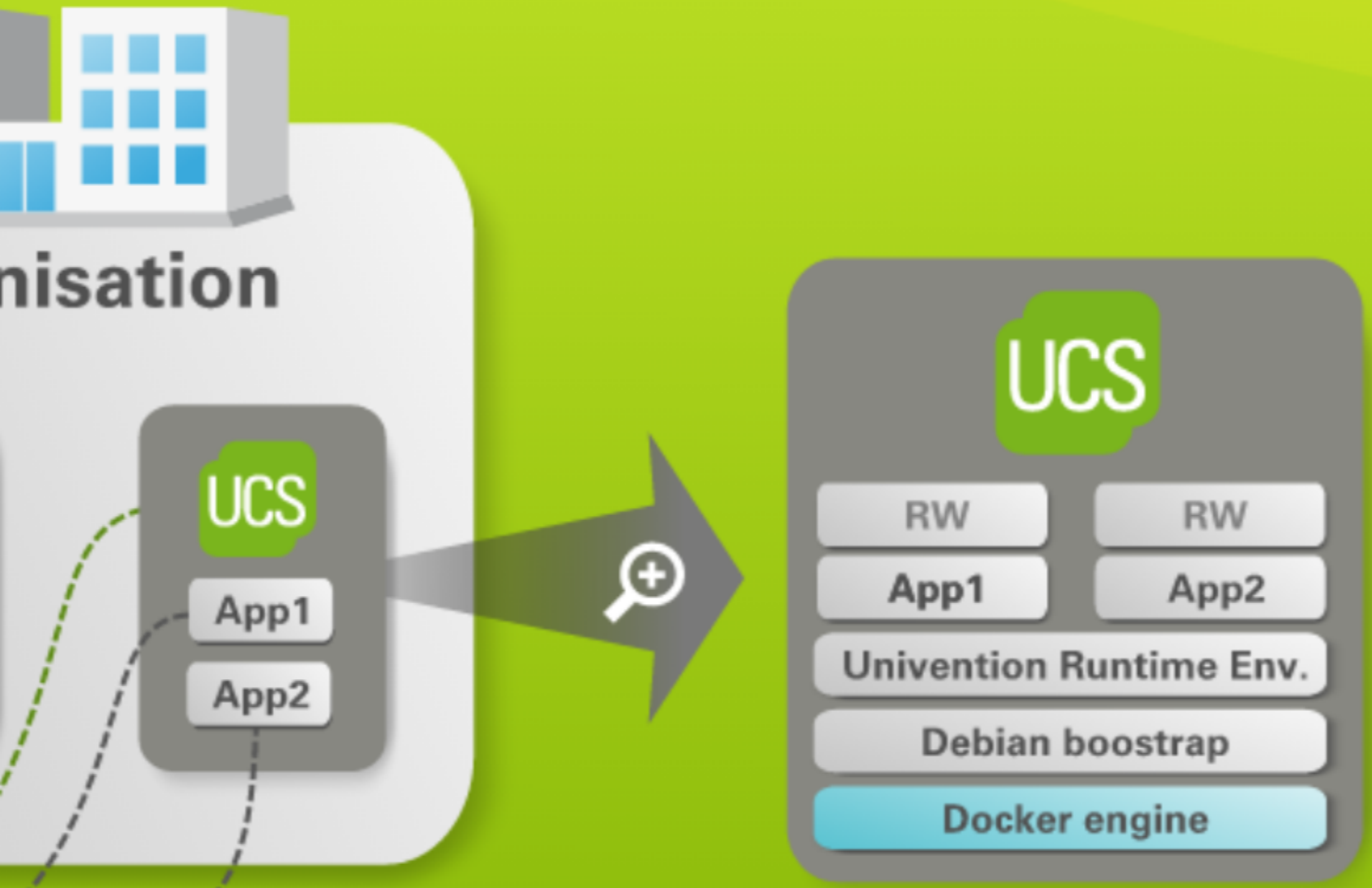
A Platform for Running Secured 3rd Party Server Applications

Dr. Alexander Kläser
klaeser@univention.de
Open Source Software Engineer



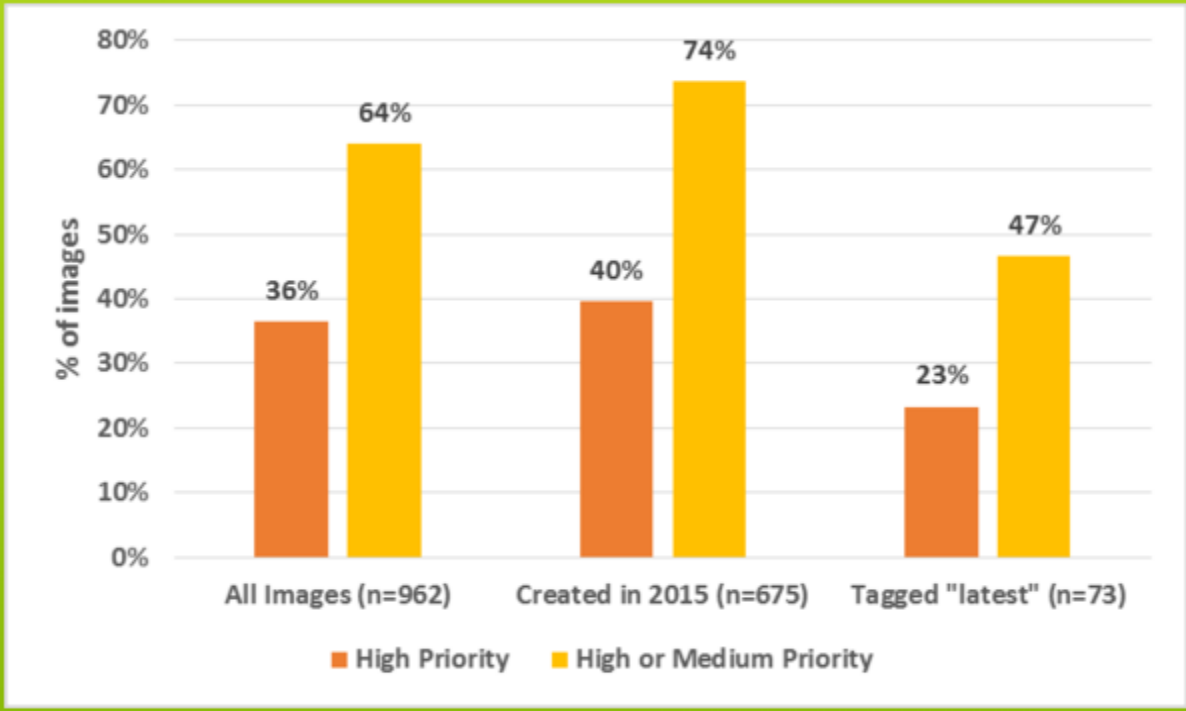

Our way
towards Docker


Organisation
UCS
Domain Controller Master
UCS
App1
App2
UCS
RW
RW
App1
App2
Univention Runtime Env.
Debian boostrap
Docker engine
UCS Domain
UCS
App1
App2
UCS
App1
App2


nisation
UCS
App1
App2
UCS
RW
RW
App1
App2
Univention Runtime Env.
Debian boostrap
Docker engine

Docker and security...

Over 30% of official images in Docker Hub contain high priority security vulnerabilities (05/2015)



http://www.banyanops.com/blog/analyzing-docker-hub/

How to update UCS based containers?

- Existing Debian packaged UCS apps
  - Run Debian update routines within container
  - Works out of the box
  - Updated packages are not shared among containers
- Native Docker containers
  - Discard container and get an updated one
  - Extensible migration logic to persist data via dedicated scripts

How to keep data persistent during updates?

- Via dedicated mount point */var/lib/univention-appcenter*
- Mount point exists on host + in container
- Preferred location for configuration and user data
- Useful for migration step during container updates

Safer Apps?

- Research project together with the German Research Center for Artificial Intelligence (DFKI)
- Funded by National Ministry of Economy and Technology (BMWi)
- Goal: How can apps be run in a secure manner?
  - Ensure the security of the IT infrastructure and other apps (w.r.t. confidentiality + integrity)
  - Transparently show consequences of an app installation to the user

Proposed solution for *Safer Apps*

- Container technology (i.e., Docker) for sandboxing
  - Integrate SELinux/AppAmor to harden container security (Mandatory Access Control = MAC)
- Abstract and define resource access for each app (files/directories, network, LDAP directory)
  - Allows to analyze information flow and infer security implications
  - Show security implications to user (during installation)
  - Automatically enforce security policies (MAC, LDAP, firewall)

Browser security

- App needs user credentials to verify user and to access LDAP information
- Direct login at app web interface is a security risk
- Single Sign-On (SSO) protocols (e.g., SAML) are ideal
- ... but not all apps support it

Proposed solution: Pseudo SSO process

- User authenticates himself at identity provider
- ... and chooses an app to access
- One-time password is generated for the app
- User is automatically logged in at and forwarded to the app

Roadmap

- Support for non-UCS based containers is productive **:-)**
- First containerized apps have been published **:-)**
- UCS apps will automatically be published at Docker Hub, Amazon, Azure, Google
- ... in addition to downloadable images (KVM, VirtualBox, VMWare, Hyper-V)
- Existing apps will be migrated into containers
- Conventions for container apps will be refined
- Support for multiple containers per app?
- *Safer Apps* prototype implementation

Thank you very much for your attention!

- More information about UCS:
  http://www.univention.com
- Cost free edition available for download!
- Documentation and developer guide:
  http://docs.univention.de/

- HTML5 presentation created with:
  https://github.com/alexklaeser/impressive.js