

Grenzen überwinden mit dem neuen Desktop-as-a-Service DESIGN

CCX Cloud Computing Digital Conference 2024

Prof. Dr. Christian Baun

Frankfurt University of Applied Sciences
(1971–2014: Fachhochschule Frankfurt am Main)
Fachbereich Informatik und Ingenieurwissenschaften
christianbaun@fb2.fra-uas.de



Schwierige Entscheidungen... wer kennt sie nicht?!



Gäbe es doch nur eine Lösung, die alles bietet...

Eine echte Herausforderung unserer Zeit. . .

Lizenzen aller Bilder: CC0

Entscheidung, welche Bereitstellungsstrategie die beste ist, fühlt sich manchmal an wie die Entscheidung zwischen. . .



**Lokal bereitgestellte
Anwendungen**

(„native Linux/Windows-
Anwendungen“)

oder. . .



+



**Software-as-a-Service –
SaaS**

(„Web-Anwendungen“)

Vorteile und Nachteile traditioneller IT vs. SaaS

Lokal bereitgestellte Anwendungen

(„native Linux/Windows-Anwendungen“)

- Datenschutz und Datensicherheit
- Unterstützung von Legacy-Anwendungen
- Kein Vendor Lock-in möglich
- Client erfordert bestimmtes OS
- Client muss vertrauenswürdig sein
- Aufwand für Administration
- Fernzugriff ist schwierig
- Keine automatische Synchronisation
- Keine automatische Datensicherung

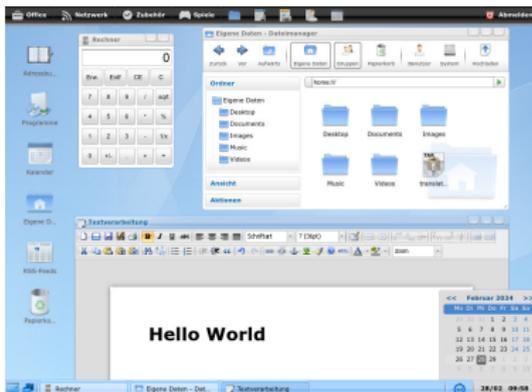
Software-as-a-Service – SaaS

(„Web-Anwendungen“)

- Jeder Client mit Browser ist geeignet
- Synchronisation und Backups
- Kein Einrichtungsaufwand (Public-Cloud)
- Viel Einrichtungsaufwand (Private-Cloud)
- Anbieter muss vertrauenswürdig sein
- Risiko eines Vendor Lock-in
- Netzwerkverbindung erforderlich
- Problem: Legacy-Anwendungen
- Potenzielle Probleme bzgl. Datenschutz und Datensicherheit

Auch moderne VDI-Lösungen (Virtual Desktop Infrastruktur) wie Azure Virtual Desktop oder Windows 365 haben Nachteile. Sie erhöhen die Komplexität (weiterer Desktop), erfordern viel Konfiguration und sind in der Regel keine freie Software

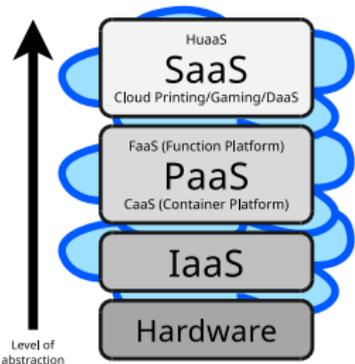
Inspiration: Web-Desktops \approx Desktop-as-a-Service



- Web-Desktops realisieren eine Art Desktop-as-a-Service (DaaS)
 - Beispiel: **eyeOS** (2005-2014)

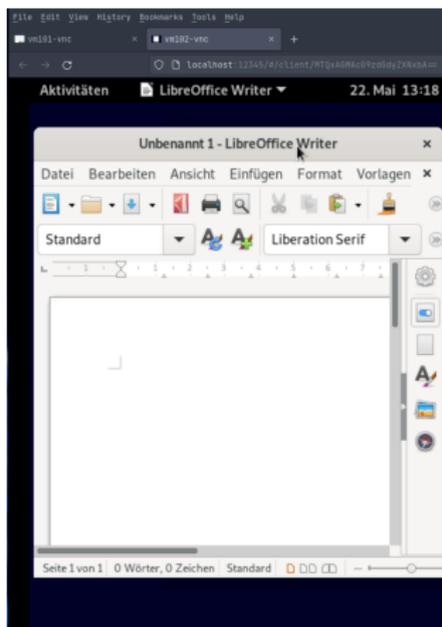
Letzte freie Version (AGPL-Lizenz): v2.5 (2011)
<https://github.com/nawawi/eyeOS>
<https://github.com/jonrandoem/eyeos>
 2014 von Telefónica aufgekauft \Rightarrow † 2017
 Nachfolgeprojekt **oneye** \Rightarrow † 2023
<https://github.com/oneye/oneye>

- Ansatz von eyeOS: Ein Desktop wird via PHP und (statischem) HTML nachgebaut bzw. „emuliert“
 - Internet-Technologien von vor 20 Jahren
 - **Gescheitertes Konzept**
 - Kein Linux/Windows-Anwendungen
 - Dennoch eine Lösung, die inspirieren kann:
 - Benutzer benötigen nur einen Browser
 - Lauffähig in Public Clouds sowie on-premises
 - Freie Software & kein Vendor Lock-in



Motivation zur Entwicklung eines neuartigen DaaS

Kann man ein System oder eine Dienstleistung entwickeln, das/die alle genannten Vorteile bietet und die Nachteile vermeidet?



Mit der Entwicklung von DaaS-DESIGN versuchen wir das!

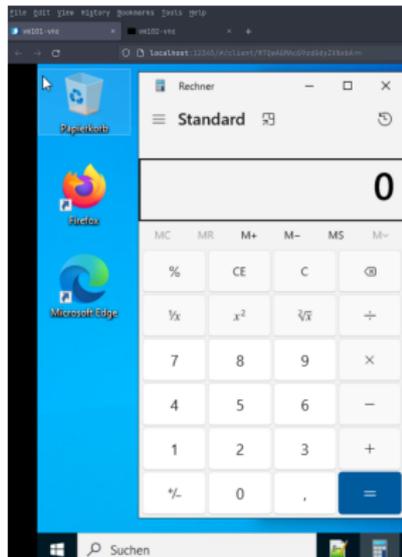


Ein moderner Desktop-as-a-Service mit unvergleichlichen Merkmalen:

- ① Integration unmodifizierter Linux/Windows-Anwendungen
- ② Fokus = Anwendungen und nicht Desktops
- ③ Interaktion ausschließlich via Browser
- ④ (Fast) ausschließlich freie Software (Open Source)
- ⑤ Bereitstellung vor Ort und Nutzung als Service möglich
- ⑥ Hohe Zuverlässigkeit durch Cluster-betrieb
- ⑦ Keine Probleme mit Datensicherheit, Datenschutz und Vendor Lock-in

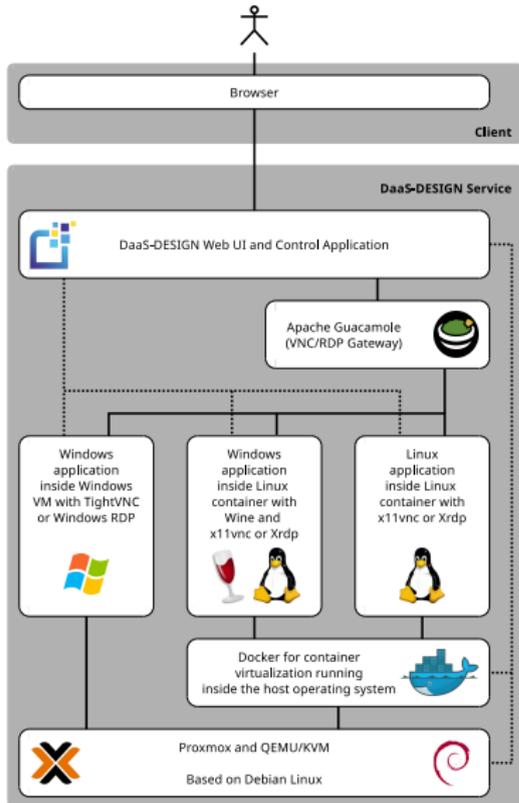
Konsequenzen für die Architektur

Die angestrebten Eigenschaften des Gesamtsystems bestimmen die Anforderungen an die zu integrierenden Komponenten



- **Linux und Windows-Anwendungen betreiben**
 - ⇒ Container und virtuelle Maschinen sind essenziell und verbessern durch isolierten Betrieb die Sicherheit
- **Fernsteuerung** der Linux/Windows-Anwendungen
 - ⇒ Container/VMs brauchen einen VNC- oder RDP-Dienst
- **Einzelne Anwendungen exportieren.** Keine Desktops
 - ⇒ Wenige VNC-/RDP-Server ermöglichen das (u.a. x11vnc, TightVNC, Xrdp, RDP in Windows)
- **Interaktionen komplett via Browser**
 - ⇒ Ein Gateway übersetzt die GUI in einen Webstream (Apache Guacamole)
- **Lokalen und entfernten Betrieb ermöglichen**
 - ⇒ Voraussetzung: Etablierte freie Lösungen (u.a. Proxmox Server-Virtualisierungsplattform)
 - ⇒ Gute Voraussetzungen für hohes Datenschutz-Niveau und kein Risiko für einen Vendor Lock-in

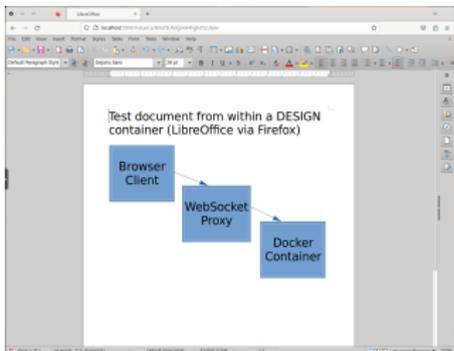
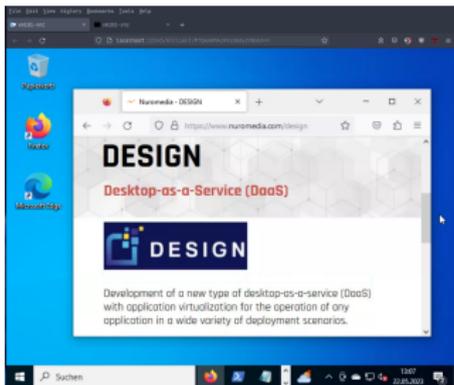
Architektur



- Apache Guacamole als VNC/RDP Gateway
- **3 Bereitstellungsarten für Anwendungen:**
 - 1 Linux-Anwendungen in Linux-Containern
 - 2 Windows-Anwendungen in Linux-Containern mit Wine (wenn es funktioniert)
 - 3 Windows-Anwendungen in Windows-VMs (wenn Wine nicht funktioniert)
- Die Proxmox Server-Virtualisierungsplattform bietet u.a. VMs (KVM), Container (LXC), Speicher, Netzwerke, Multi-Node Deployment, ...
- Docker stellt eine API und eine Reihe von Funktionen bereit, die LXC überlegen sind

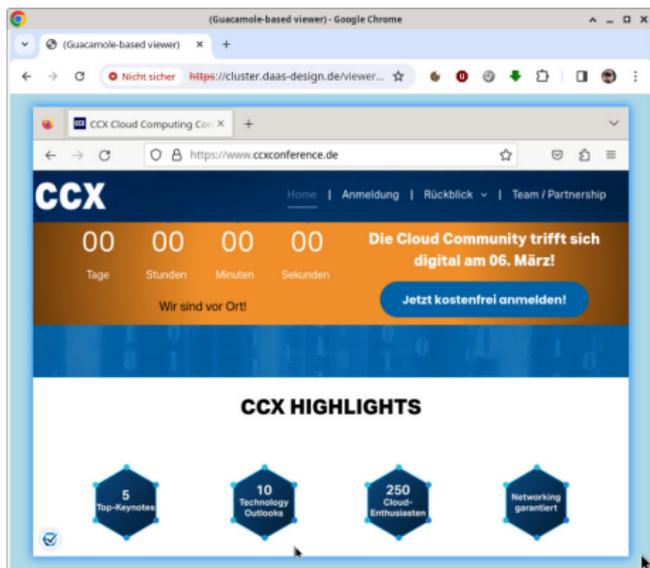
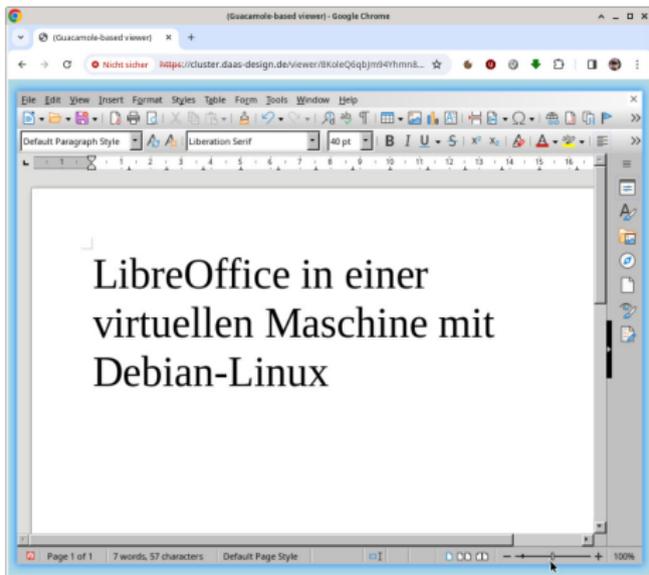
Alle Komponenten (außer das Windows-Betriebssystem innerhalb der Windows-VMs) sind freie Software (Open Source)

Einige (teilweise noch ungelöste) Herausforderungen

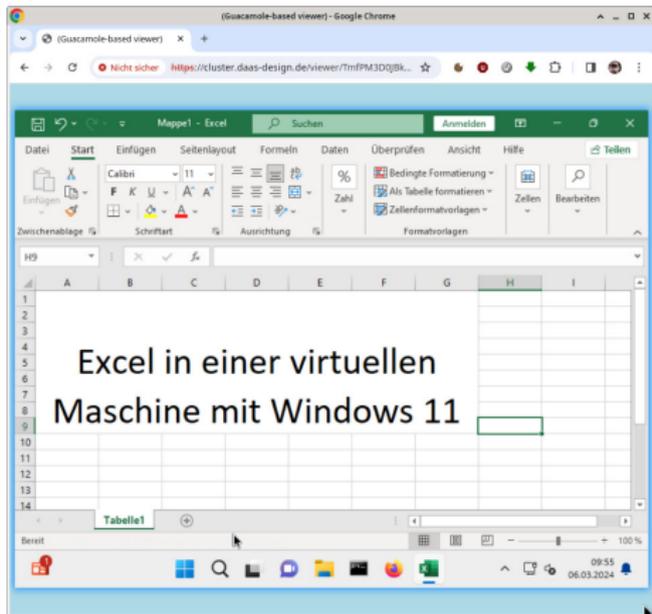
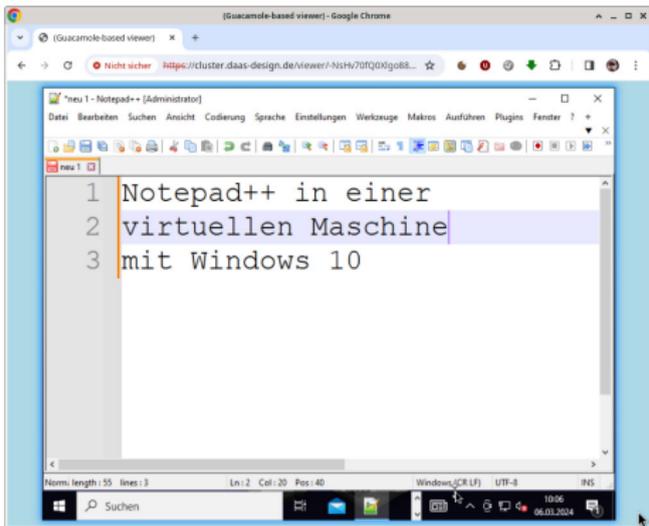


- Die GUI einer Anwendung via VNC oder RDP aus einem Linux-Container exportieren ist trivial
 - Aus Windows-Containern ist es aber unmöglich
 - Nicht alle Windows-Anwendungen laufen mit Wine
 - ⇒ Verlangt eine Anwendung eine echte Windows Betriebssystemumgebung, sind eine VM und **Windows-Lizenz** nötig
- Anwendungen laufen isoliert in Containern und VMs
 - **Datenaustausch** muss dennoch möglich sein
 - ⇒ Proxmox bietet das verteilte Dateisystem Ceph. Es muss aber in Container und VMs automatisch integriert werden
 - Offene Fragen: **Drucken, Sound, Mikrofon, . . .**
- Virtualisierung und Gateway haben Auswirkungen
 - Ist die **Bereitstellungszeit** von Anwendungen ok?
 - Sind **Performance** und **Latenz** der Interaktion akzeptabel für alle Anwendungen?

Prototyp: Aktueller Stand der Entwicklung (1/2)



Prototyp: Aktueller Stand der Entwicklung (2/2)



Danke für Ihre Aufmerksamkeit! Fragen?



- DESIGN ist eine gemeinsame Entwicklung von Nuromedia und der Frankfurt University of Applied Sciences, gefördert durch das Bundesministerium für Wirtschaft und Klimaschutz
- Verfügbar am Herbst/Winter 2024
- Eine URL, die man sich merken sollte:
<https://www.daas-design.de>
- Fragen? \implies christianbaun@fb2.fra-uas.de

Weitere Infos zur Architektur und Komponentenauswahl

Christian Baun, Johannes Bouché.

Closing the Gap between Web Applications and Desktop Applications by designing a novel Desktop-as-a-Service (DaaS) with seamless support for Desktop Applications.

Open Journal of Cloud Computing (OJCC), 8(1):1–19, 2023

