

# ChatOps

=

## Kommunikation + Kollaboration + DevOps

# Max Ferstl



# Christian Kühn

# Agenda

Kommunikation

ChatOps - was ist das?

ChatOps mit Hubot  
(aus unserem Arbeitsalltag)

Samples

# Kommunikation

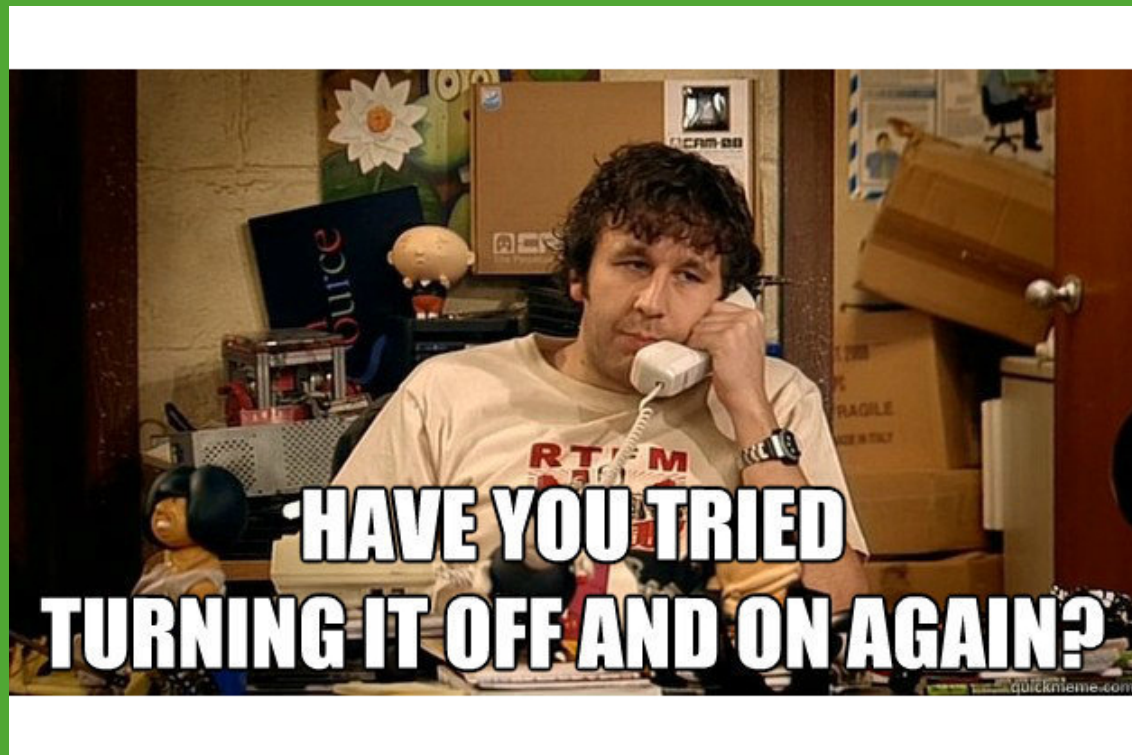
*(lat. communicatio, ‚Mitteilung‘) ist der Austausch oder die Übertragung von Informationen.*

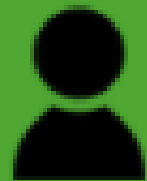
*„Information“ ist in diesem Zusammenhang eine zusammenfassende Bezeichnung für Wissen, Erkenntnis oder Erfahrung.*

<https://de.wikipedia.org/wiki/Kommunikation>

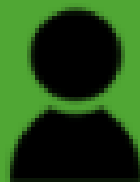
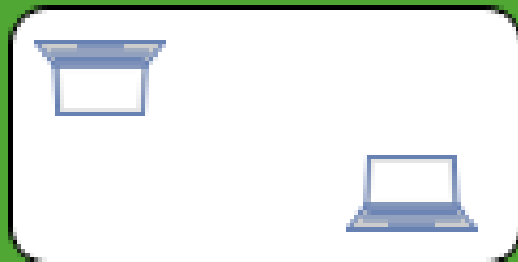
# Kommunikation (klassisch IT / OPS)

Telefon  
Mail  
Ticket



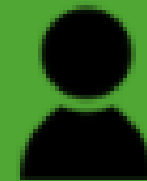


chris

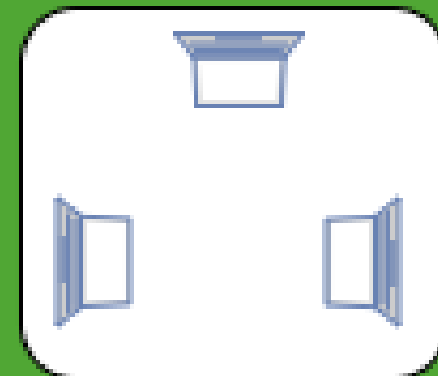


Max

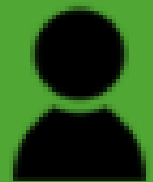
**Ops**



John  
Software-Developer

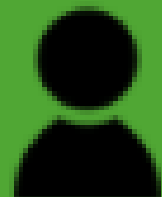


Maria  
Kunde PO



Hans  
DevOps

**Kunde  
cust1**



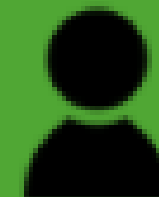
chris



"Anfrage"

*Ausführung*

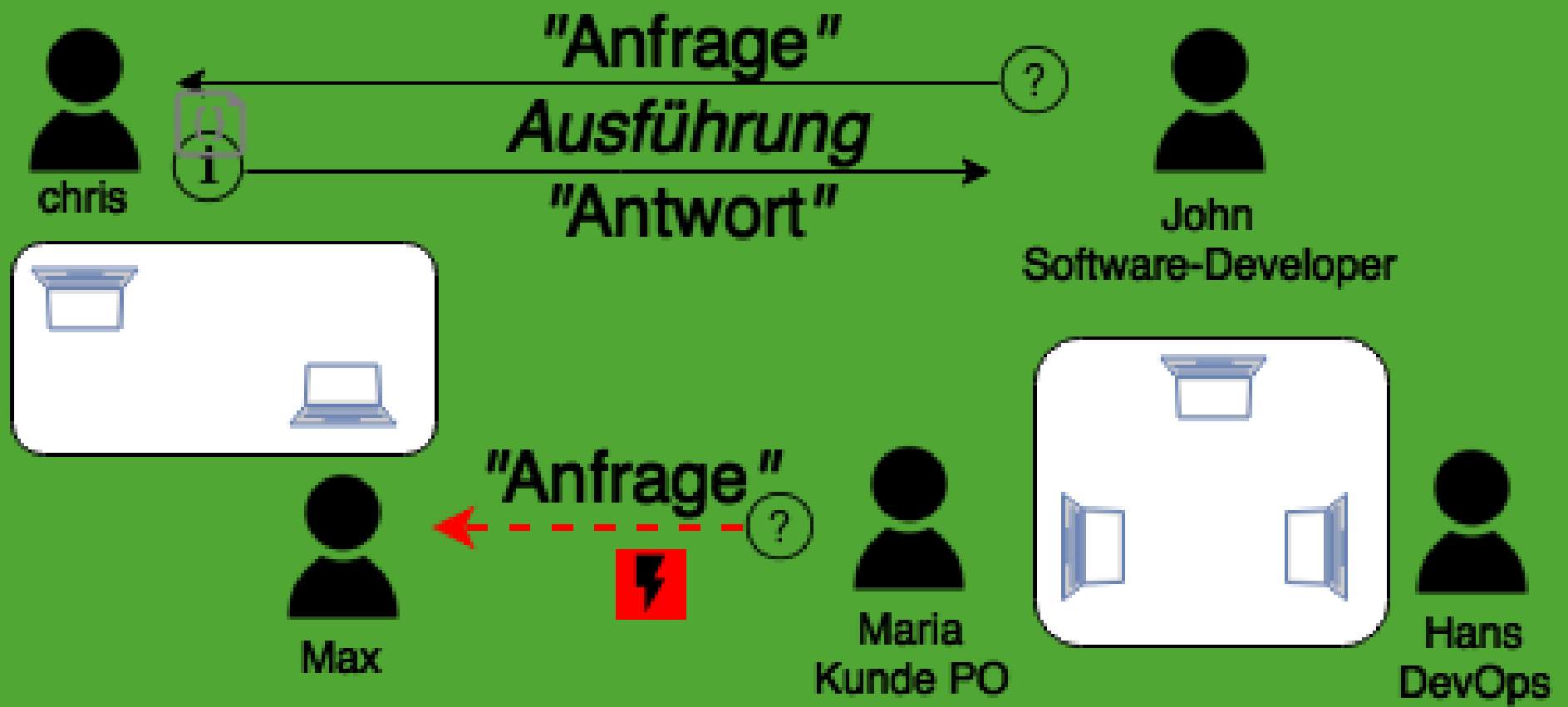
"Antwort"



John

Software-Developer







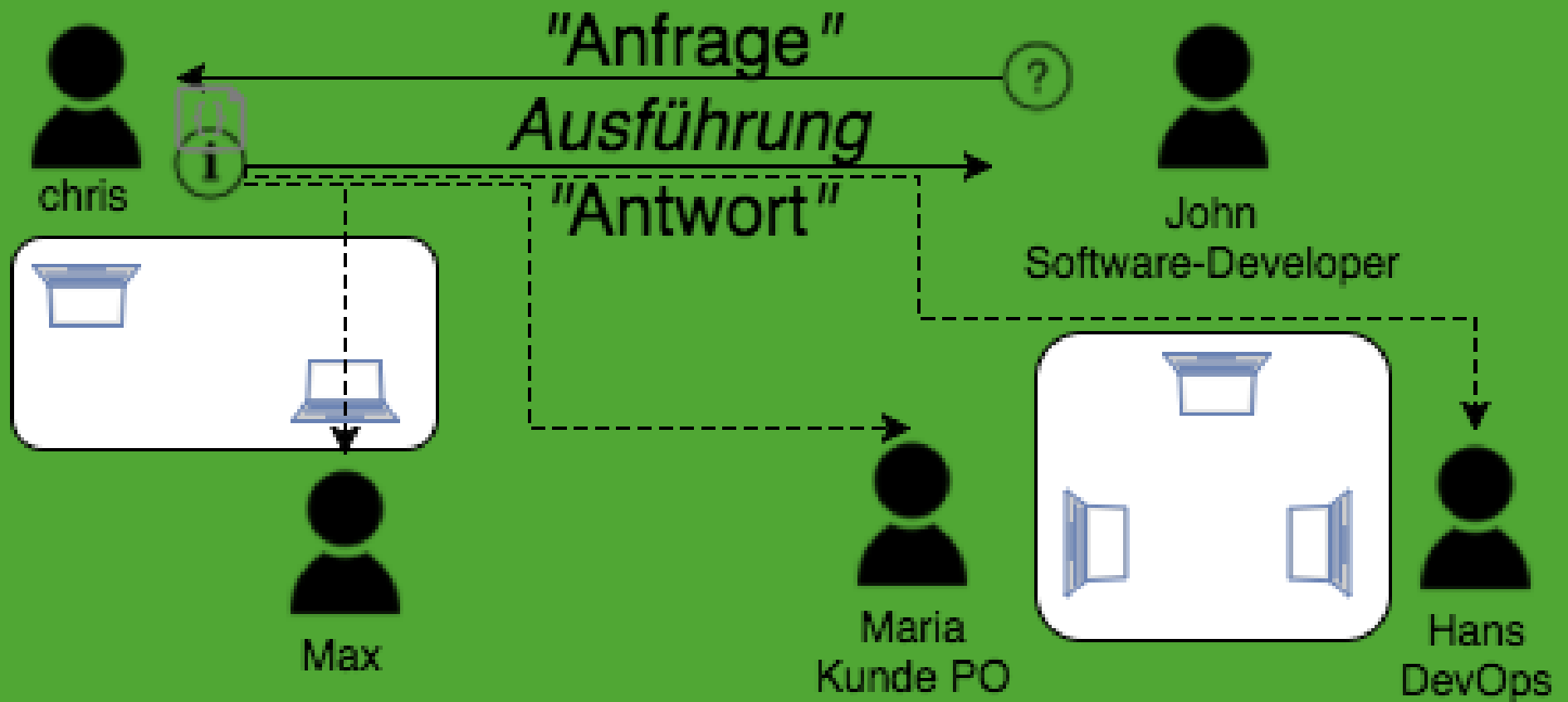
# Kommunikation (für Teams)

IRC

Slack | Hipchat | Rocketchat

Skype | Hangout

XMPP/Jabber



```
13:17      john : hallo liebe admins, ich brauch einen Account fuer das git-repo "cust1"
13:17      chris : ok, ich verbind mich kurz zum git host und mach das
13:18      chris : john: habs eingerichtet.
13:18      maria : ach toll chris hat ja den account schon angelegt
```

Kommunikation

Zeitersparnis

Transparenz

**=> chatbot als middleware**

## Ausführung des Befehls

gleichzeitige Doku ohne Extraaufwand

```
13:41      john : hallo liebe admins, ich brauch einen Account fuer das git-repo "cust1"  
13:41      chris : demobot: gitlab add john:cust1  
13:41      demobot : ✓ Added john to gitlab group 'cust1'!  
13:41      maria : danke
```

**Transparenz**

**WER**

**hat WANN**

**WAS ausgeführt**

WER - chris

hat WANN - 13:41

WAS ausgeführt - *gitlab add john:cust1*"

```
13:41      john : hallo liebe admins, ich brauch einen Account fuer das git-repo "cust1"  
13:41      chris : demobot: gitlab add john:cust1  
13:41      demobot : ✓ Added john to gitlab group 'cust1'!  
13:41      maria : danke
```

# **shared shell**

ersetzt die private script-Sammlung

schnelle Verteilung/Verbreitung von scripten



# Befehle sind:

unabhängig vom Benutzersystem

unabhängig vom Zielsystem

MacOSX, Linux, Windows, Android, BSD, Embedded/Appliances...

# Vereinfachte Ausführung

syntax bildet nur echte use cases ab

Bot bzw. shell reagiert nur auf eingeschränktes Befehlsset

## zentrale Ausführung

Wissen über alle Server und Dienste nicht so wichtig

# Wissensverteilung, Training

Lernen durch mitlesen

erkennen "neuer" Befehle und Skripte

```
john : hallo liebe admins, ich brauch einen Account fuer das git-repo "cust1"  
chris : demobot: gitlab add john:cust1  
demobot : ✓ Added john to gitlab group 'cust1'!  
maria : danke  
cy : .oO(oh cool so legt man also gitlab-accounts an)
```

# Befehle auf die User verlagern

Anfrage triggert sofortige Bearbeitung inkl. Antwort und Doku

```
john : hallo liebe admins, ich brauch einen Account fuer das git-repo "cust1"  
chris : demobot: gitlab add john:cust1  
demobot : ✓ Added john to gitlab group 'cust1'!  
maria : hey das sieht ja einfach aus, koennte ich das in zukunft auch selbst einrichten?  
maria : demobot: gitlab add hans:cust1
```

# Sicherheit

bot ist sehr mächtig

```
maria : hey das sieht ja einfach aus, koennte ich das auch selbst in zukunft einrichten?  
maria : demobot: gitlab add hans:cust1  
demobot : maria: X Not for you, anyway. no such user online or nickname not registered  
maria : demobot: gitlab add hans:cust1
```

=> Einschränkung per ACL

=> Einschränkung per bot-Konfiguration

## Security: ACL

Anmeldung am Bot direkt (per private message / query)

Anmeldung am Chat (z.b. bei Slack, HipChat)

ACL über chat-nickname (z.b. Nickserv)

## Security: bot-Konfiguration

pro Team eine Bot-Instanz

gemeinsame Codebase (Wissenstransfer etc)

Teambot lauscht nur im chatroom #team

Konfiguration beinhaltet nur Team-eigene (API-)keys und Zugänge

## Befehle auf die User verlagern (Teil 2)

```
chris : demobot: gitlab add maria:cust1-admins
demobot : ✓ Added maria to gitlab group 'cust1-admins'!
maria : danke chris
maria : demobot: gitlab add hans:cust1
demobot : ✓ Added hans to gitlab group 'cust1'!
```



# Security: Vision

SSH-daemon abschalten

Zugriffe (nur) durch den Bot auf APIs

# Notification Management

Informationen und Alerts von aussen per webhook

z.B. <http://demobot.synyx.local:13337/demobot/jenkins-notify?room=demo>

```
demobot : [Jenkins] INFO build status for build cust1:#34683: phase COMPLETED, status SUCCESS
```

```
demobot : [Jenkins] INFO build status for build cust1:#34684: phase UNKNOWN, status FAILED
```

```
demobot : [Nagios] ERROR Host gitlab.example.com down
```

# Implementation

## Beispiele

- Hubot (JS, coffeescript)
- Lita (Ruby)
- Eggdrop (Tcl)
- Slack chat integration

# Hubot

```
npm install -g generator-hubot  
yo hubot
```

<https://github.com/github/hubot>

# hubot-scripts

<https://www.npmjs.com/search?q=hubot>

"1927 results"

```
npm install hubot-<script>
```

```
npm install hubot-audit
```

<https://github.com/github/hubot-scripts>

# Hubot synyx

"standard"-Hubot

+coffee-scripte

+bash-scripte

Kontinuierliche Erweiterung nach Bedarf

## Beispiel JS/coffeescript:

```
robot.hear /^sod\?/, (res) ->
  sod = robot.brain.get('admin:sod')
  if !sod || sod == ''
    res.send "Nobody seems to be SoD today, bin ich halt SoD"
  if /verschdl/.test(sod)
    res.send "zum Glück ist heute der Verschdl SoD, dann macht er nix kaputt"
  else
    res.send "Heute ist der #{sod} SoD und macht euch eure Rechner flott."
```

```
john : sod?
demobot : Heute ist der cy SoD und macht eure Rechner flott.
```

# Beispiel shellscript:

```
robot.hear /^fact:\s(.*)\s(.*)/, auth: {group: 'sysadmin'}, (res) ->
  command = "hack/fact.sh #{res.match[1]} #{res.match[2]}"
  execute command, res
```

```
#!/bin/bash

# pipefail, get exit code of first error
set -uf -o pipefail

CMD="sudo facter -p $2"
SSH="/usr/bin/ssh -o PasswordAuthentication=no -o StrictHostKeyChecking=no -o ConnectTimeout=5 -i /home/hubot/.ssh/id_rsa"

$SSH "bot@$1.synyx.local" "$CMD"
```

```
cy : fact: cytest2 system_uptime
demobot : ✓ {"seconds"=>602370, "hours"=>167, "days"=>6, "uptime"=>"6 days"}
```



## **don't's**

übermäßiger Implementationsaufwand

z.B. 2 Tage coden für 5min Ersparnis pro Woche

offtopic-scripte

offtopic-chat



## Beispiele synyx

```
@      cy : lanm: new host 36:32:37:39:31:61 10.0.0.0 cytest2
demobot : Creating new host with mac 36:32:37:39:31:61
demobot : ✓ Created new host cytest2.synyx.local with ip 10.0.0.137
```

## Beispiele synyx

```
cy : vm: create cytest3 project=Infra os=CentOS
demobot : cy: Creating VM: cytest3-test with options: os=CentOS env=TEST size=small
          project=Infra
```

## Beispiele synyx

```
@      cy : puppet: deactivate cytest3-test  
demobot : ✓ removed node cytest3-test from puppetdb
```

# Beispiele synyx

neuer Mitarbeiter

- Checkliste erstellen
- Mailaccount + Alias
- LDAP-account
- Groupware/Kalender-account
- Laptop (+Tickets)
- Peripherie (+Ticket)
- Telefon (+Ticket)
- Papierkrieg (Verträge)

# Beispiele synyx (geplant)

Lagerverwaltung yolo.pl

```
chris@nyx:~/synyx/infra/hwp> ./search.pl hdmi
```

name: HDMI DVI-D 2m	count: 8	ITEM_ID: 20	LOC_ID: 9 Regal 1 Fach 2
name: mini Displayport -> HDMI	count: 4	ITEM_ID: 31	LOC_ID: 10 Regal 1 Fach 3
name: HDMI A 2 Meter kabel	count: 4	ITEM_ID: 33	LOC_ID: 9 Regal 1 Fach 2
name: HDMI over IP Broadcaster	count: 2	ITEM_ID: 68	LOC_ID: 1 Regal 0 Fach 1
name: Displayport -> HDMI	count: 1	ITEM_ID: 85	LOC_ID: 1 Regal 0 Fach 1
name: HDMI 1.3b Kabel Male Male 5m	count: 2	ITEM_ID: 88	LOC_ID: 11 Regal 1 Fach 4

**jobs.synyx.de**





**DANKE!**

**Fragen?**