

Workshop

Integration eines CrossLab Gerätes

Voraussetzungen für den Hand-On Teil:

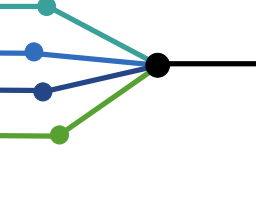
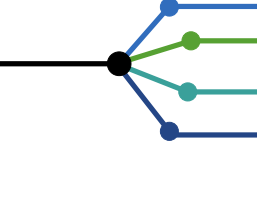
- Python 3.10 python.org/downloads
- IDE, z.B.: VS Code code.visualstudio.com
- GStreamer gstreamer.freedesktop.org/download/

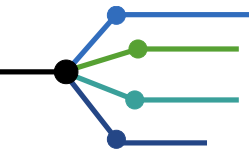
CROSS Lab



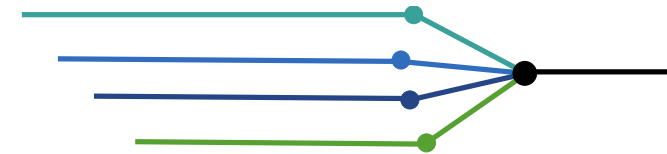
Gliederung



- Paradigmenwechsel → Verteilte (nicht monolithische) Experimente
 - Architekturkonzept
 - Integration eigener Laboratory Devices
 - Nutzungskonzept „GOLDi 2.0“
 - Hands-On: Entwicklung eines eigenen einfachen Laborgerätes
- 
- 

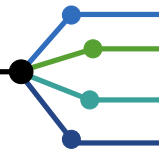
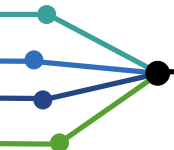
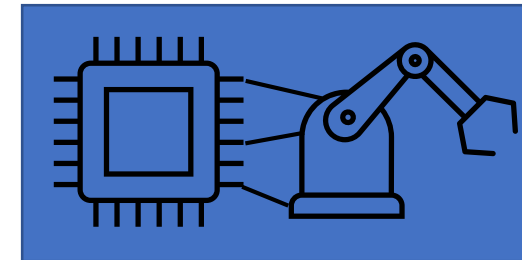


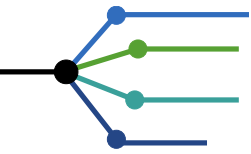
Verteilte Experimente



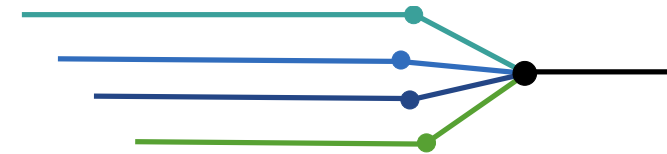
Bisher:

- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff



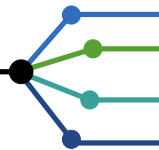
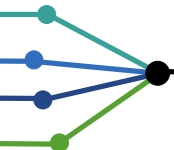
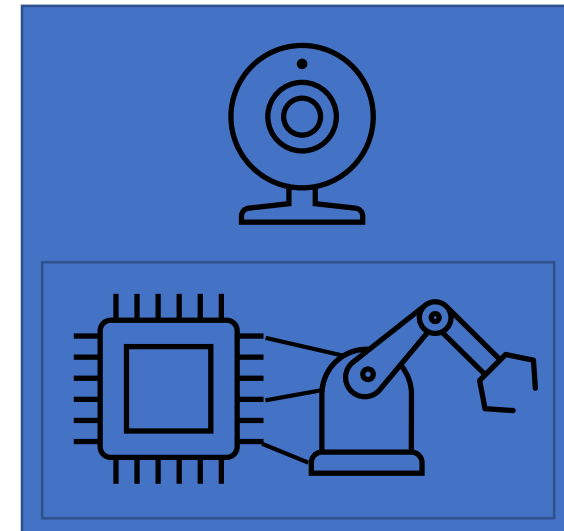


Verteilte Experimente



Bisher:

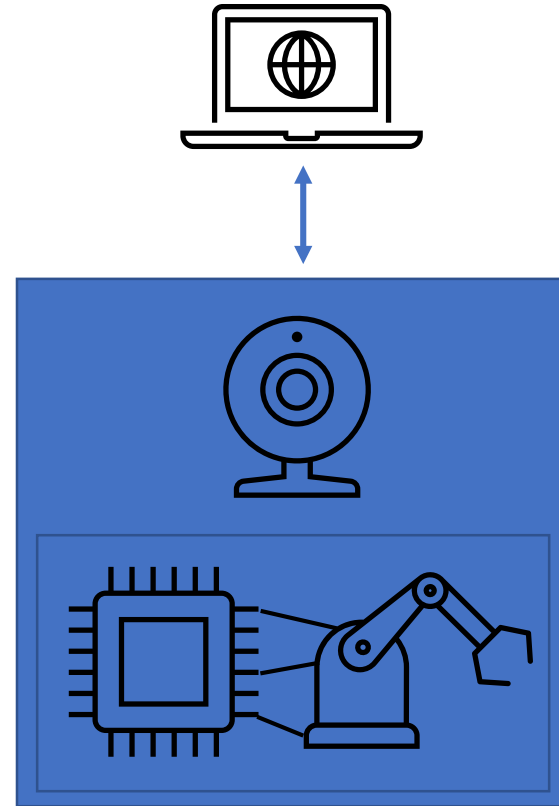
- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff



Verteilte Experimente

Bisher:

- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff



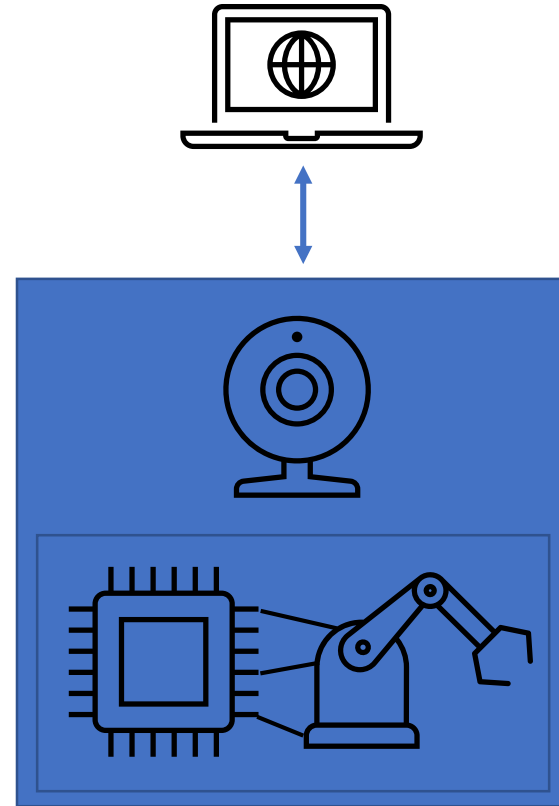
Verteilte Experimente

Bisher:

- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff

Paradigmenwechsel:

- Getrennte Laborgeräte, die sich beliebig kombinieren lassen



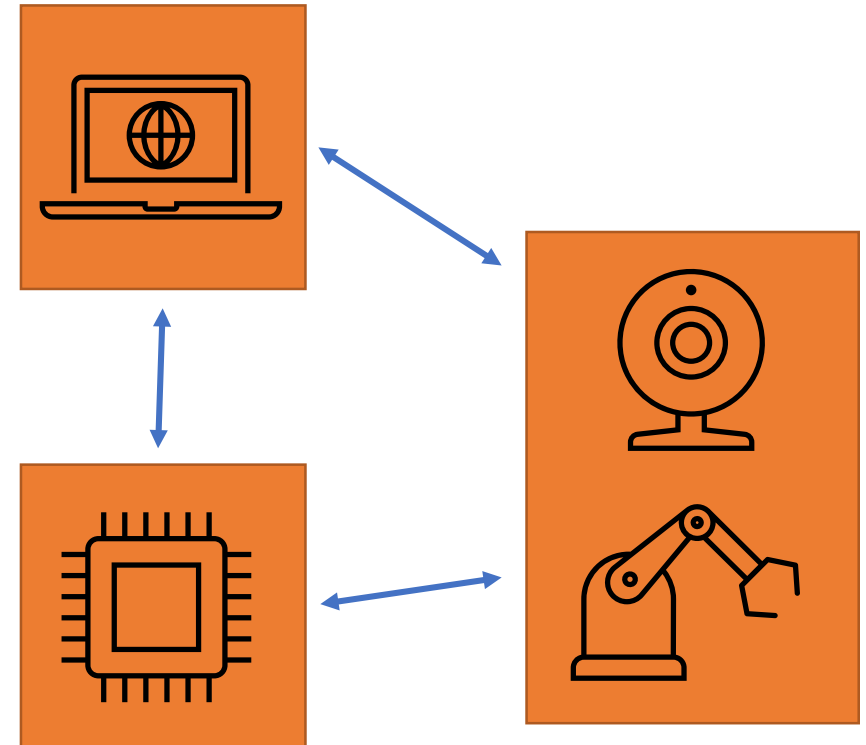
Verteilte Experimente

Bisher:

- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff

Paradigmenwechsel:

- Getrennte Laborgeräte, die sich beliebig kombinieren lassen



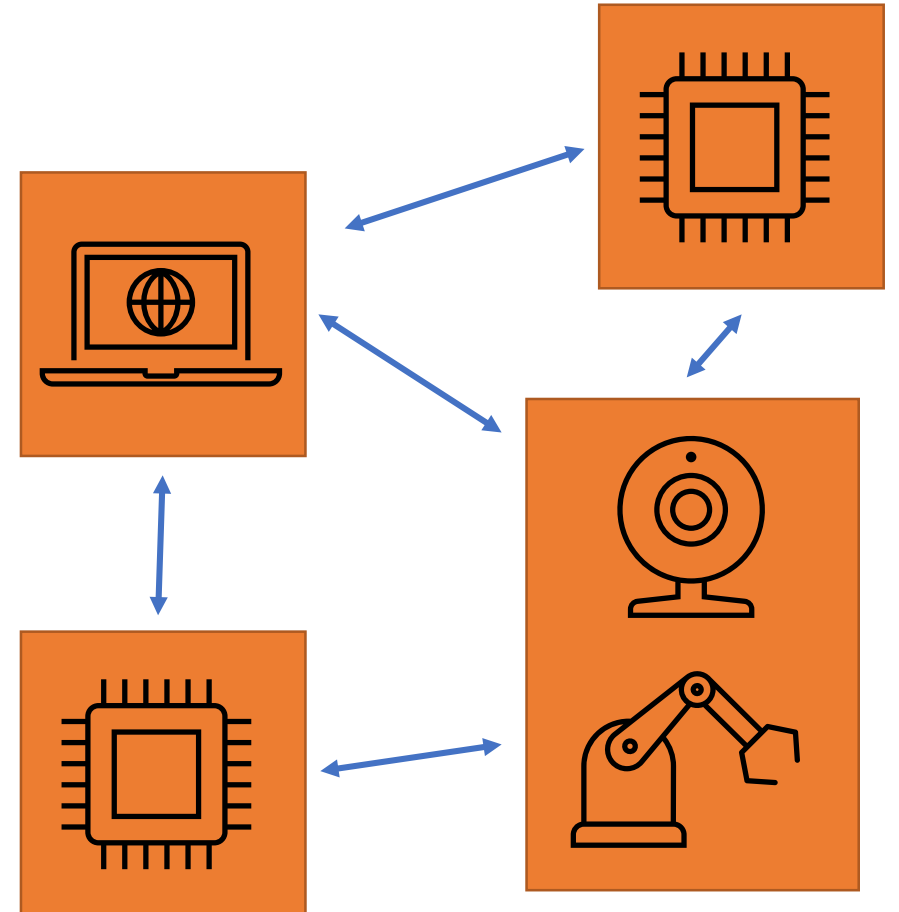
Verteilte Experimente

Bisher:

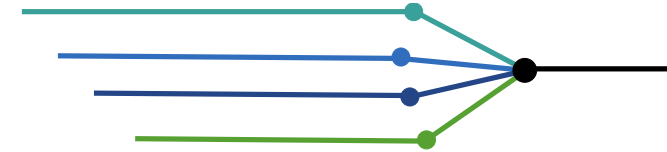
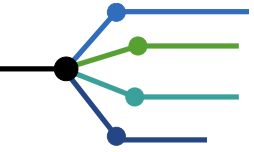
- Abgeschlossene (monolithisches) Experimente mit Web-Zugriff

Paradigmenwechsel:

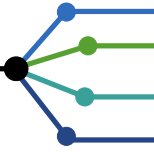
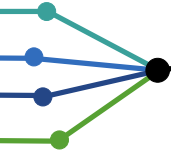
- Getrennte Laborgeräte, die sich beliebig kombinieren lassen

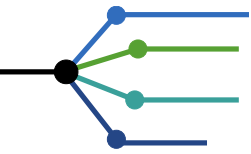


Verteilte Experimente

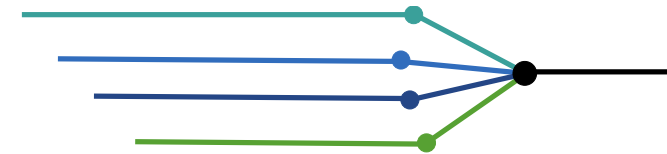


Metapher: Hands-On-Lab



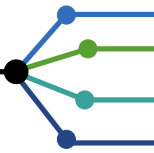
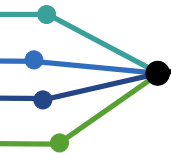


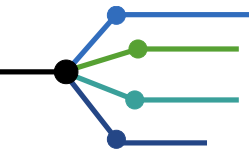
Verteilte Experimente



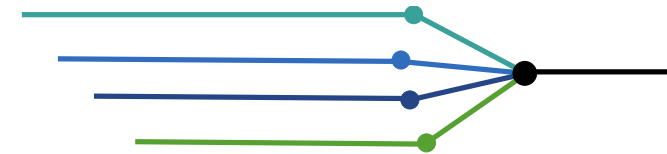
Metapher: Hands-On-Lab

Laborgerät = Laborgerät / Ausstattung





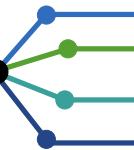
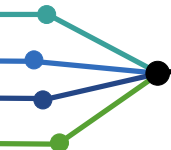
Verteilte Experimente



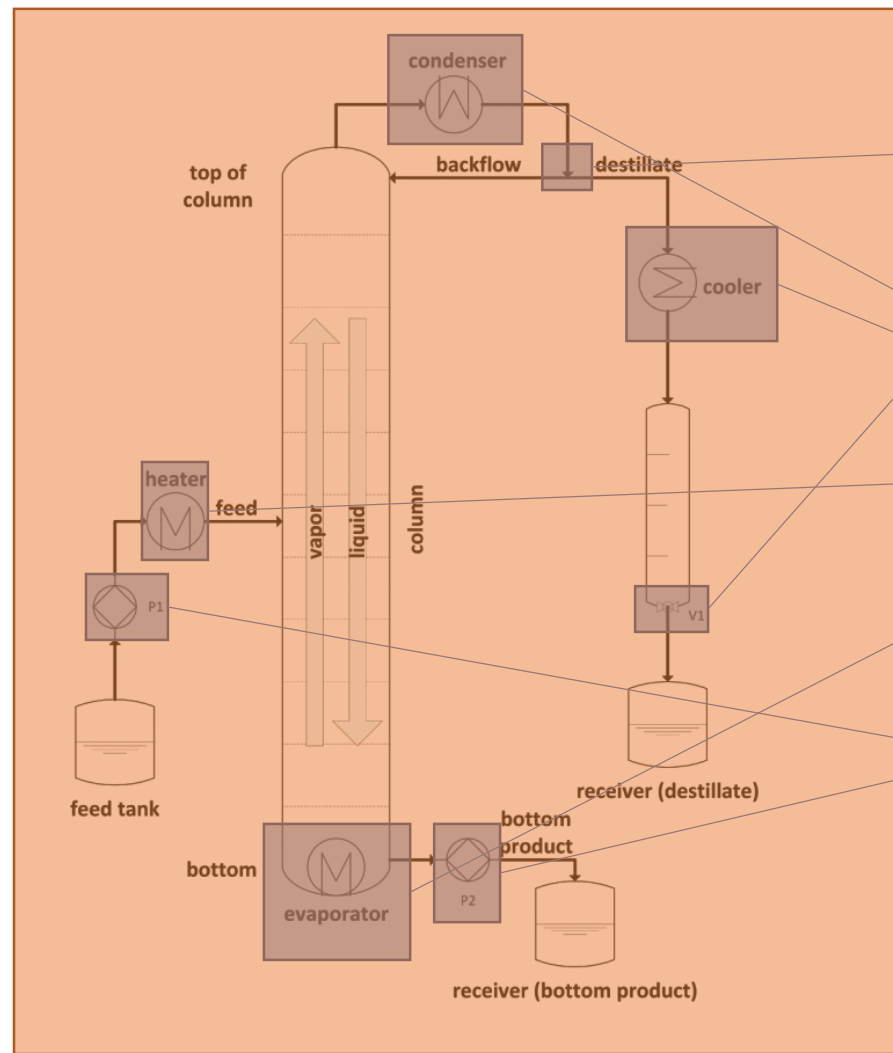
Metapher: Hands-On-Lab

Laborgerät = Laborgerät / Ausstattung

Laborelement = Atomare Funktionseinheit



Verteilte Experimente



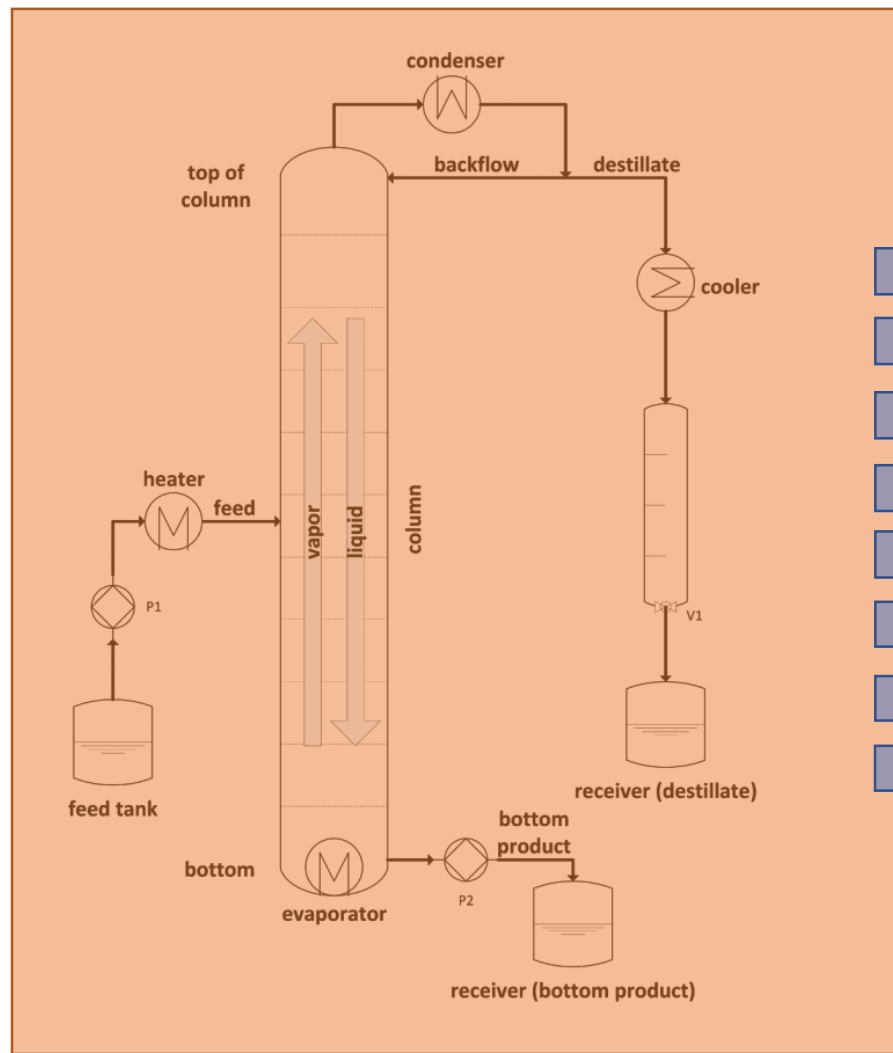
Labor Element (Ventil)

Labor Element (Heater)

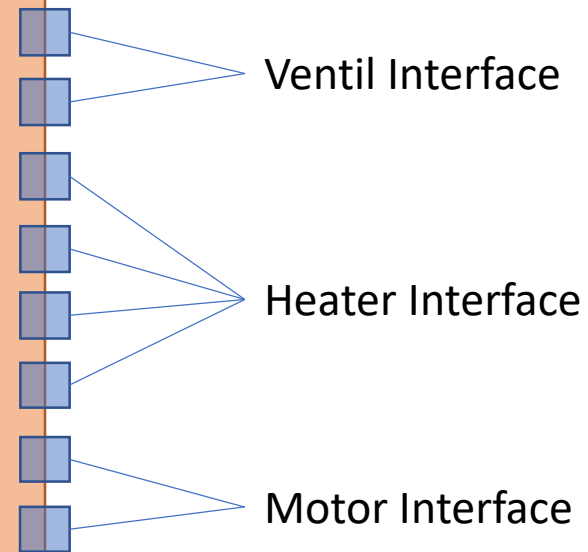
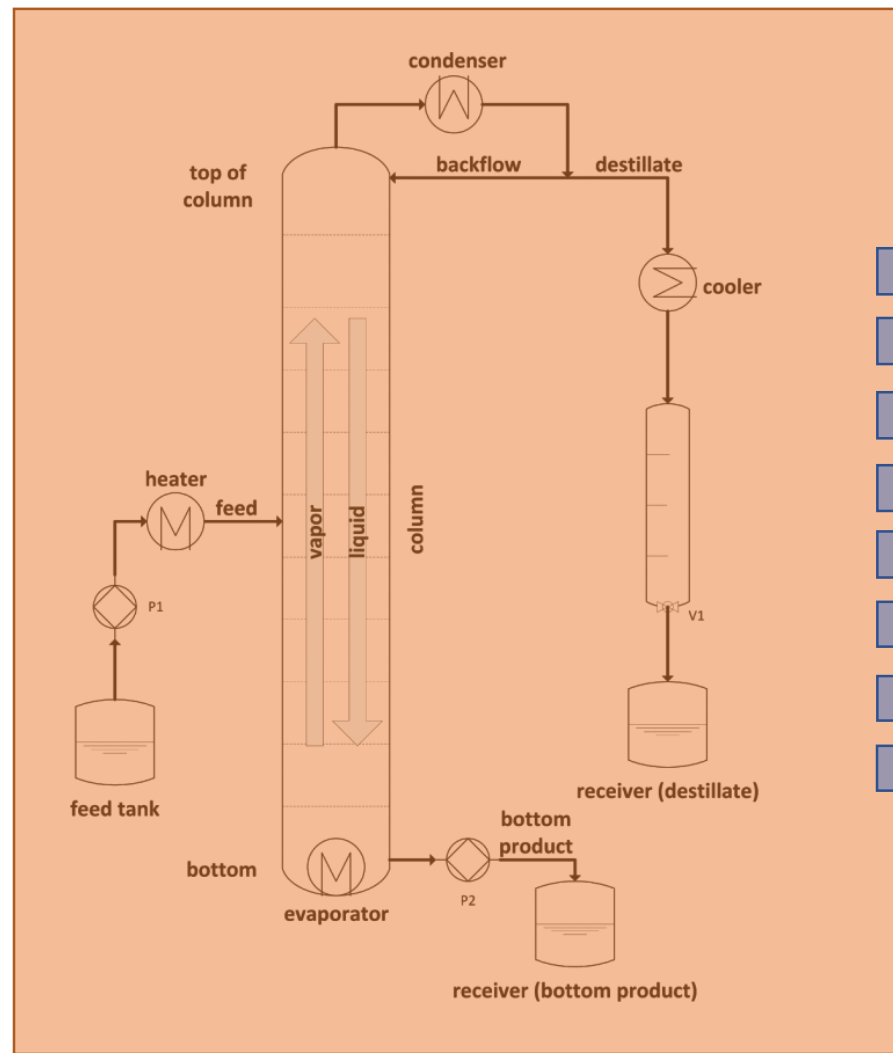
Labor Element (Motor)

Labor Gerät

Verteilte Experimente



Verteilte Experimente



Verteilte Experimente

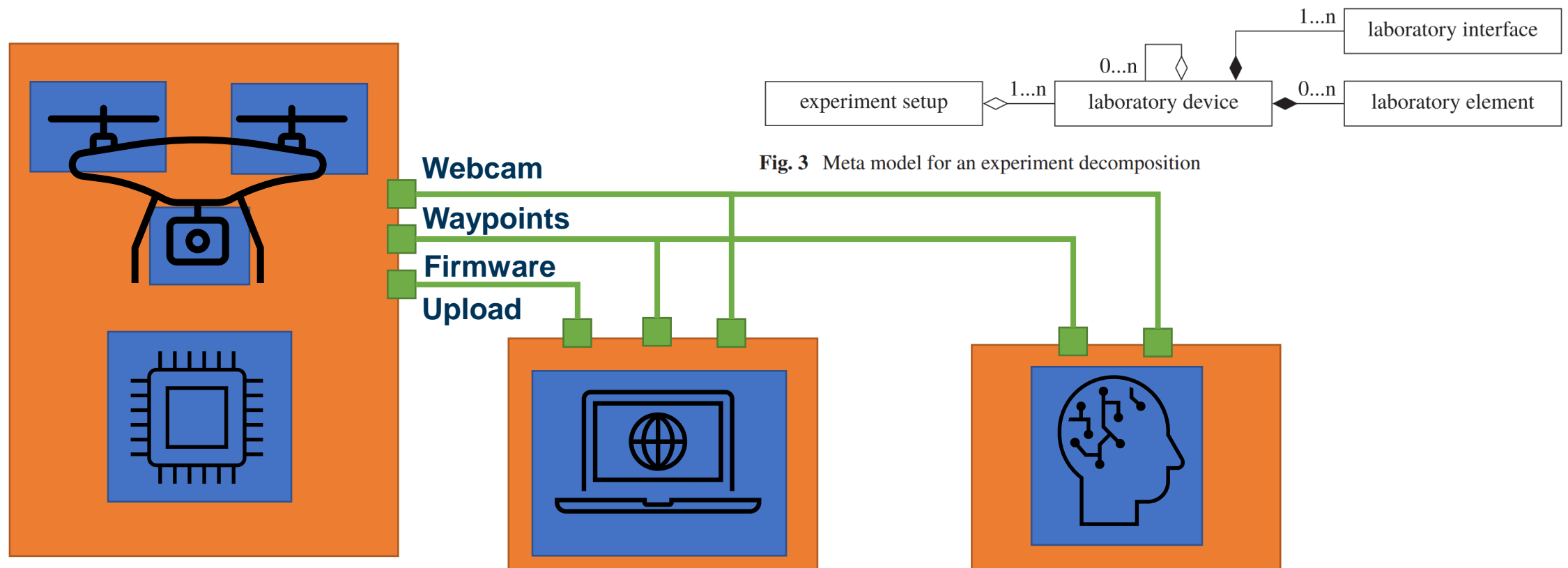
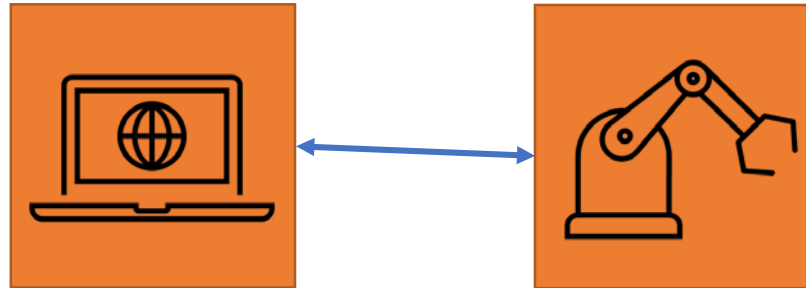


Fig. 3 Meta model for an experiment decomposition

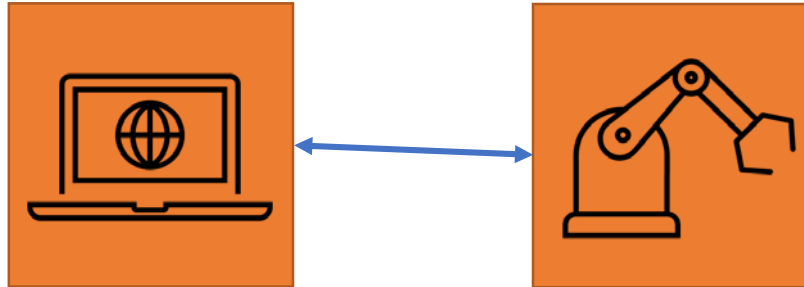
Verteilte Experimente

Monolithische Experimente

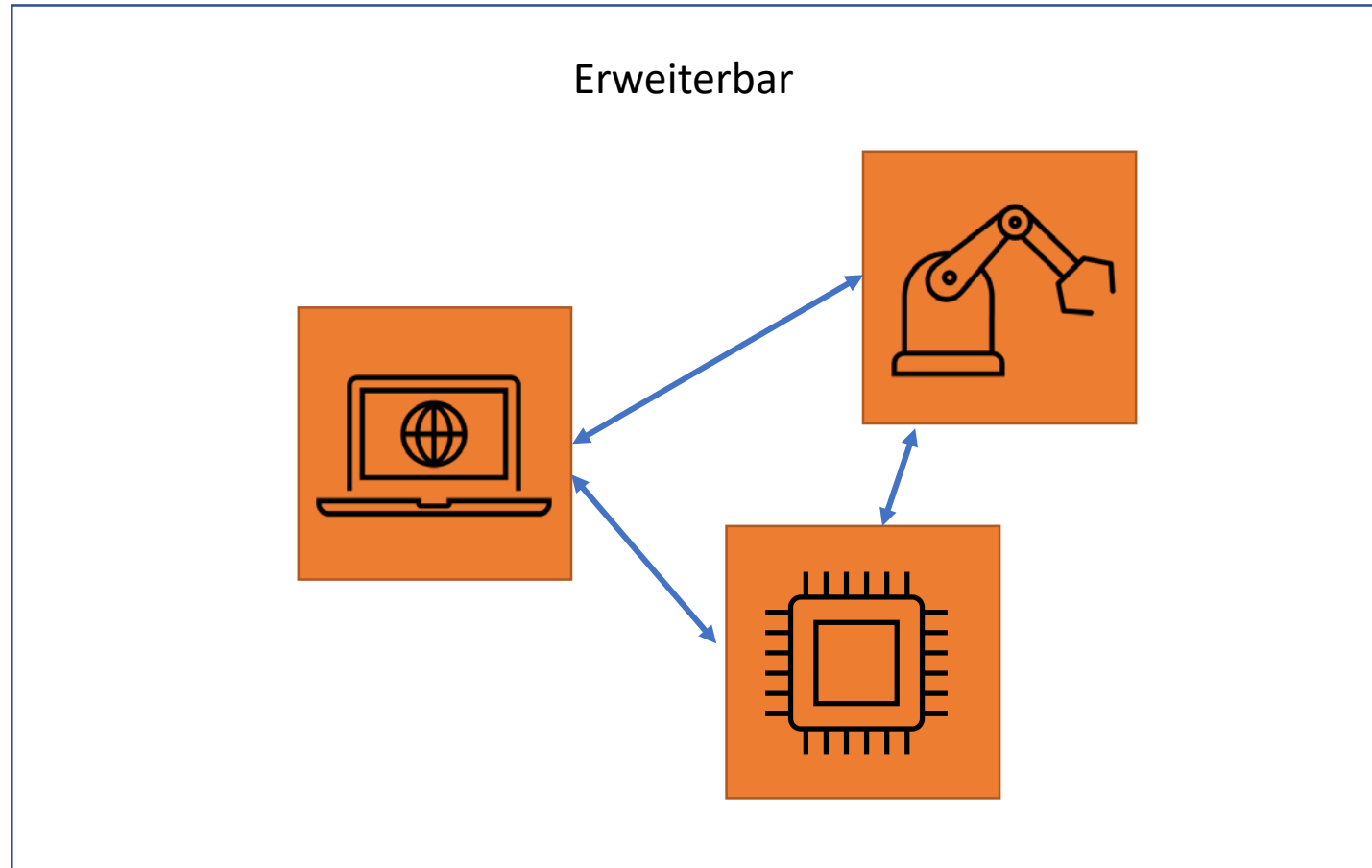


Verteilte Experimente

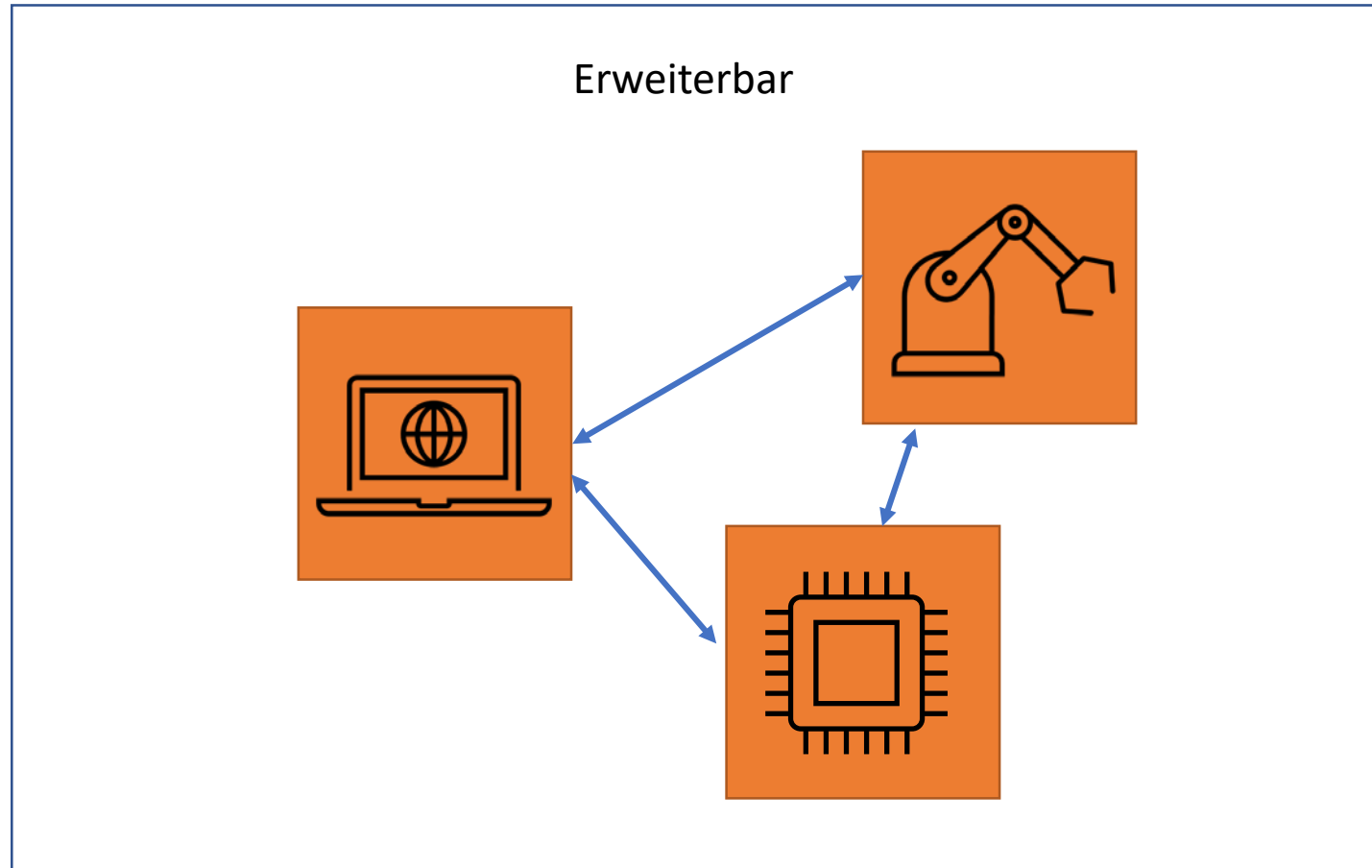
Erweiterbar



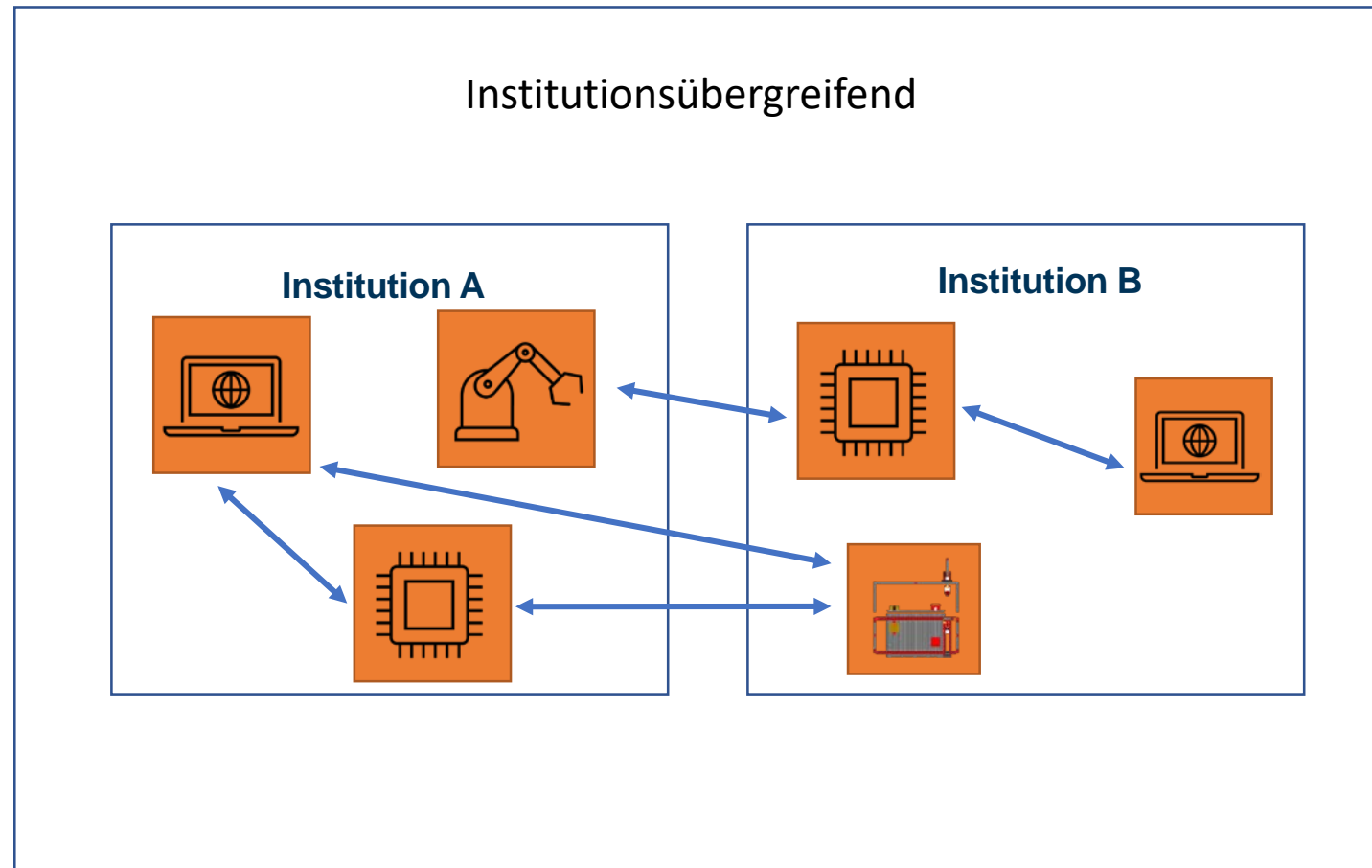
Verteilte Experimente



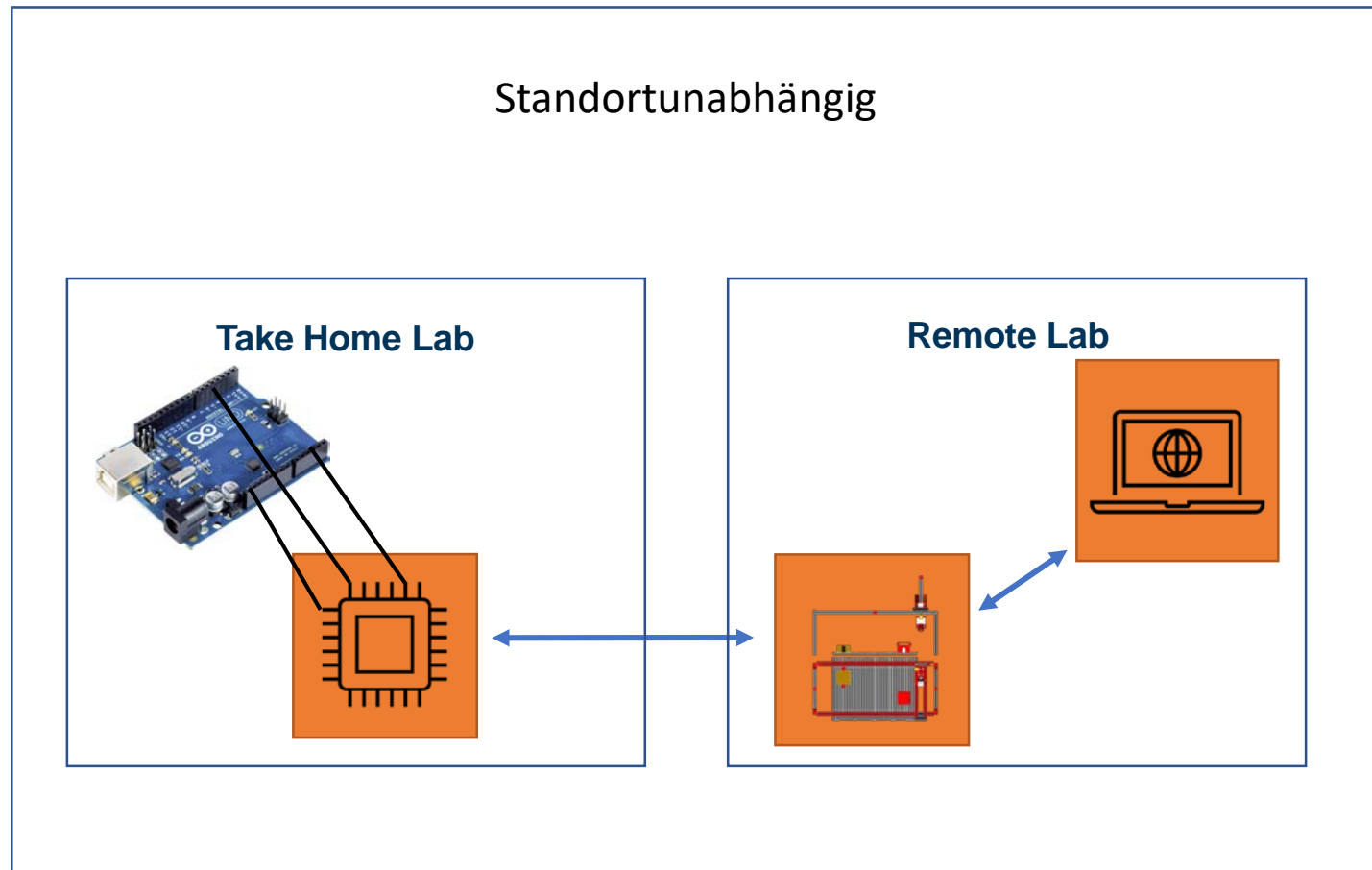
Verteilte Experimente



Verteilte Experimente

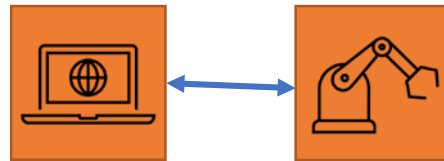


Verteilte Experimente

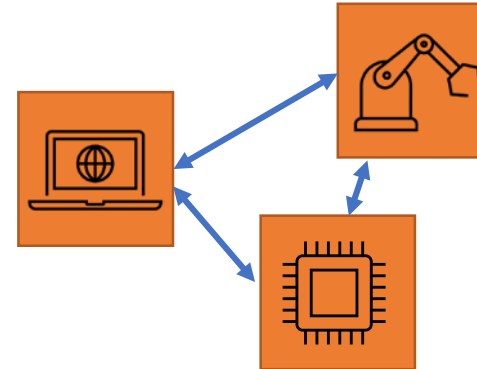


Verteilte Experimente

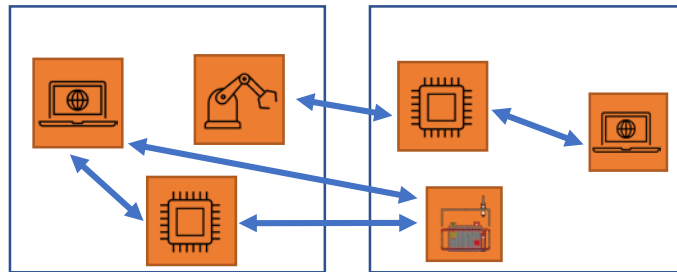
Monolithische Experimente



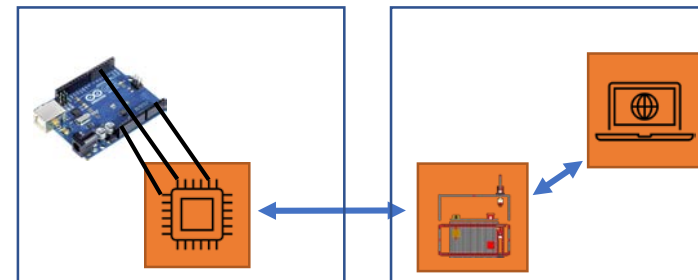
Erweiterbar



Institutionsübergreifend

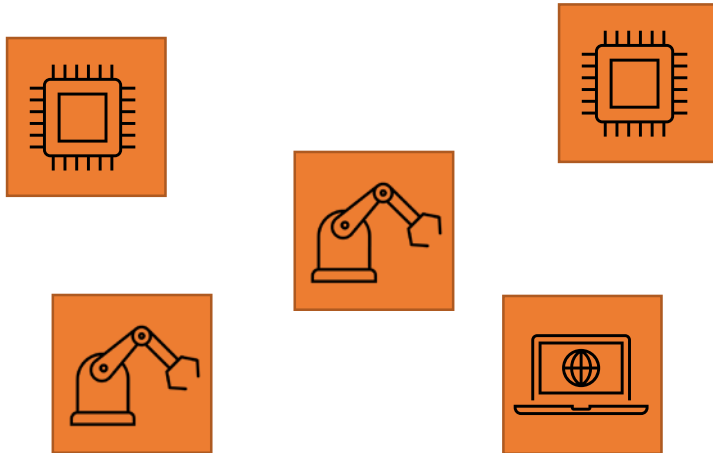


Standortunabhängig



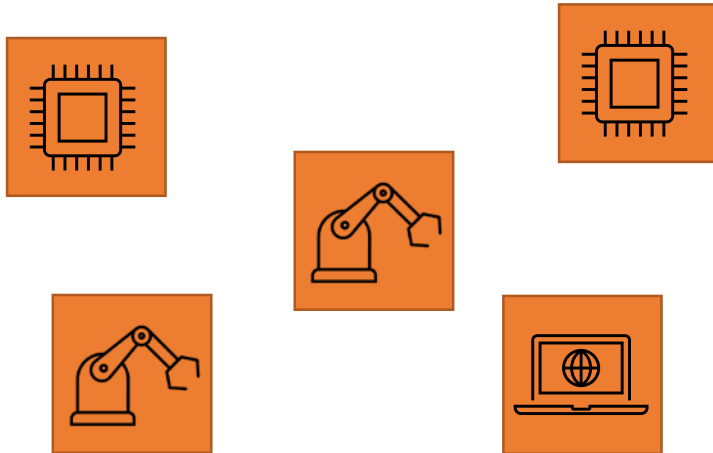
Architekturkonzept

- Laboratory Devices



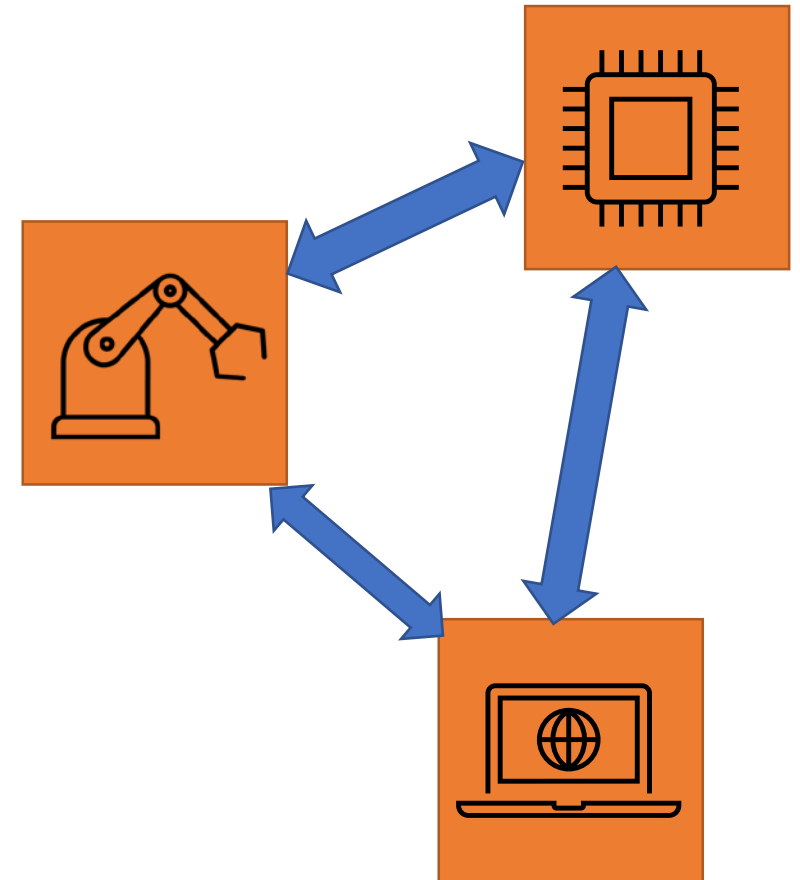
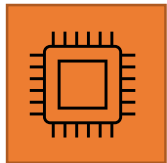
Architekturkonzept

- Laboratory Devices
- Experiment Konfiguration



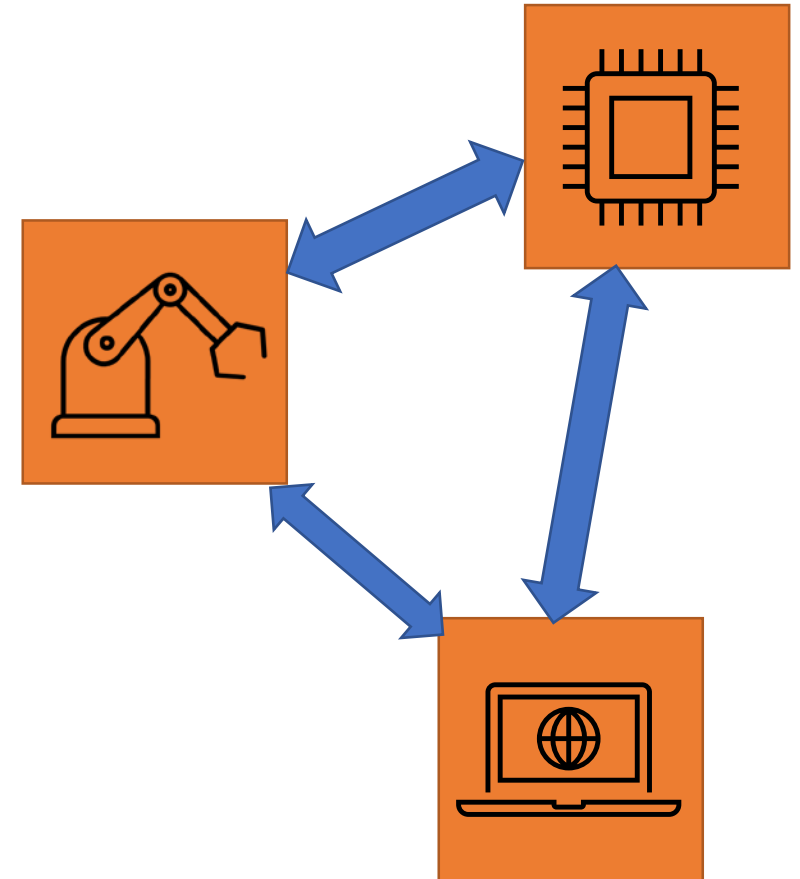
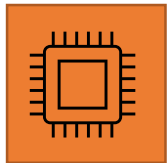
Architekturkonzept

- Laboratory Devices
- Experiment Konfiguration



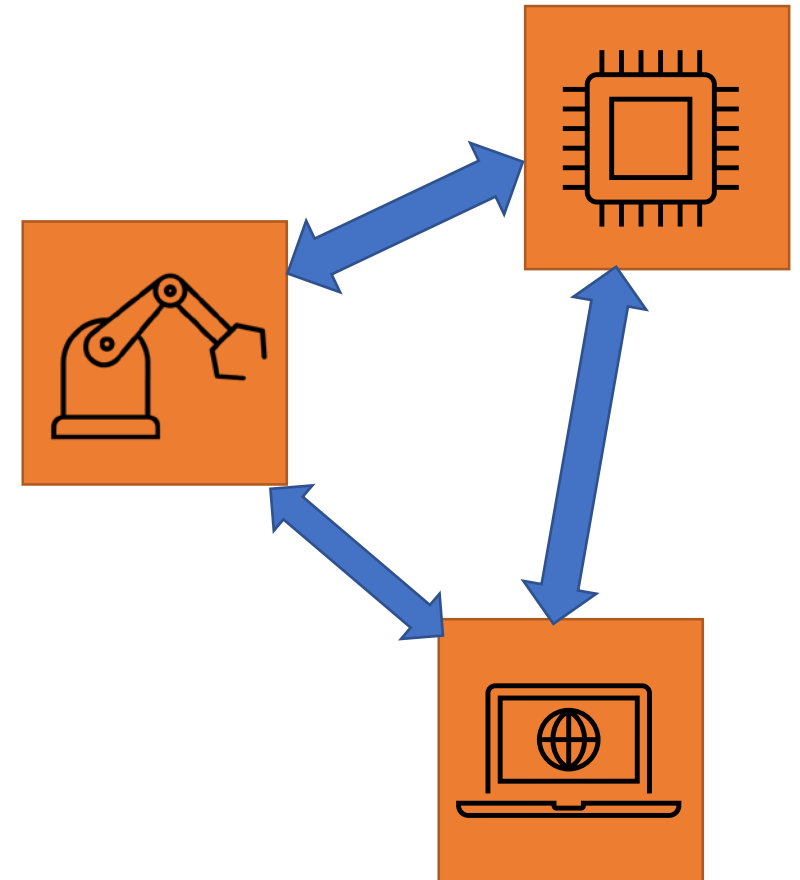
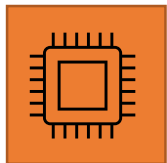
Architekturkonzept

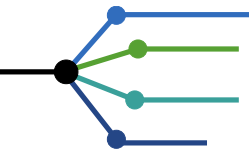
- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)



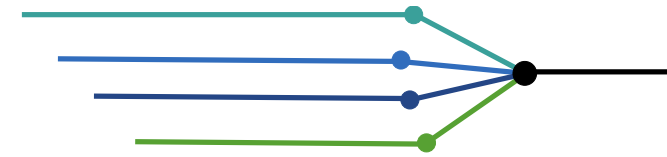
Architekturkonzept

- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments

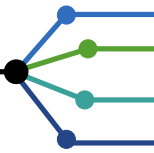
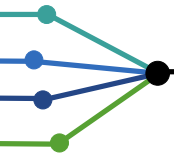




Architekturkonzept



- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments

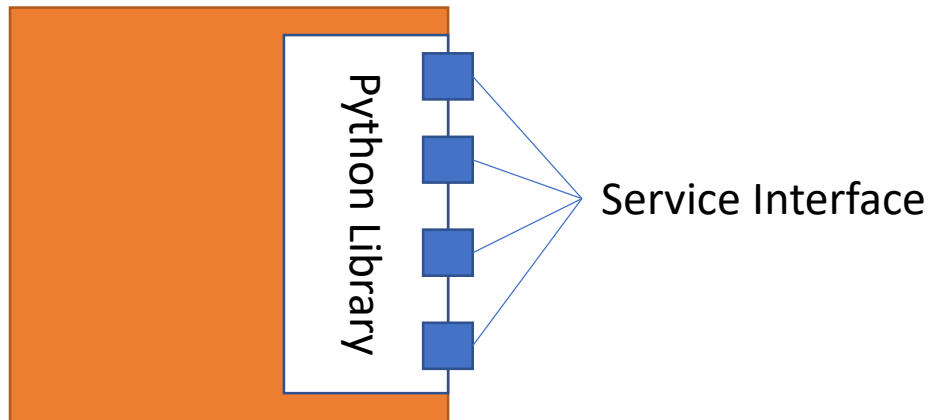




Architekturkonzept



- **Laboratory Devices**
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments

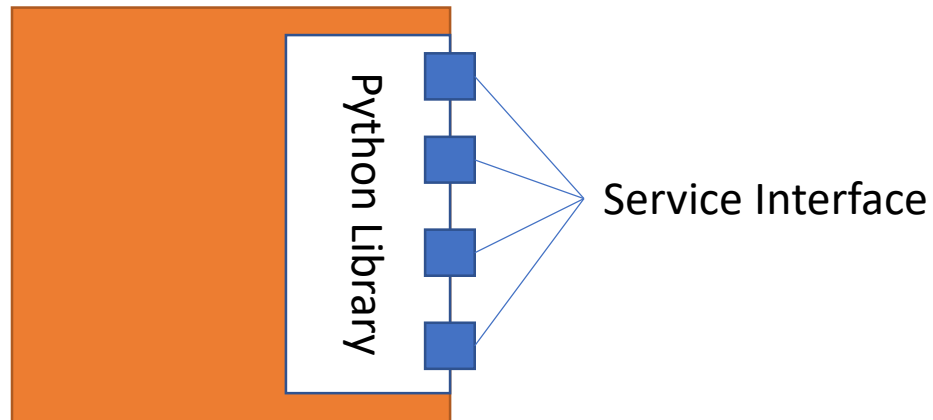




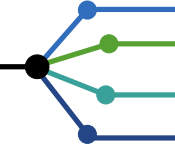
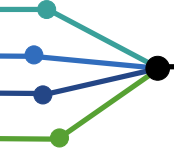
Architekturkonzept



- **Laboratory Devices**
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments



```
"name": "Quadrocopter",  
"description": "Beschreibung die das Laborgerät beschreibt",  
"services": [  
  {  
    "serviceType": "https://api.goldi-labs.de/services/Motor",  
    "serviceId": "Motor Vorne Links",  
    "serviceDirection": "in"  
  },  
  {  
    "serviceType": "https://api.goldi-labs.de/services/Motor",  
    "serviceId": "Motor Vorne Rechts",  
    "serviceDirection": "in"  
  }  
]
```



Architekturkonzept

- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments

```
"name": "Quadrocopter",  
"description": "Beschreibung die das Laborgerät beschreibt",  
"services": [  
  {  
    "serviceType": "https://api.goldi-labs.de/services/Motor",  
    "serviceId": "Motor Vorne Links",  
    "serviceDirection": "in"  
  },  
  {  
    "serviceType": "https://api.goldi-labs.de/services/Motor",  
    "serviceId": "Motor Vorne Rechts",  
    "serviceDirection": "in"  
  }  
]
```

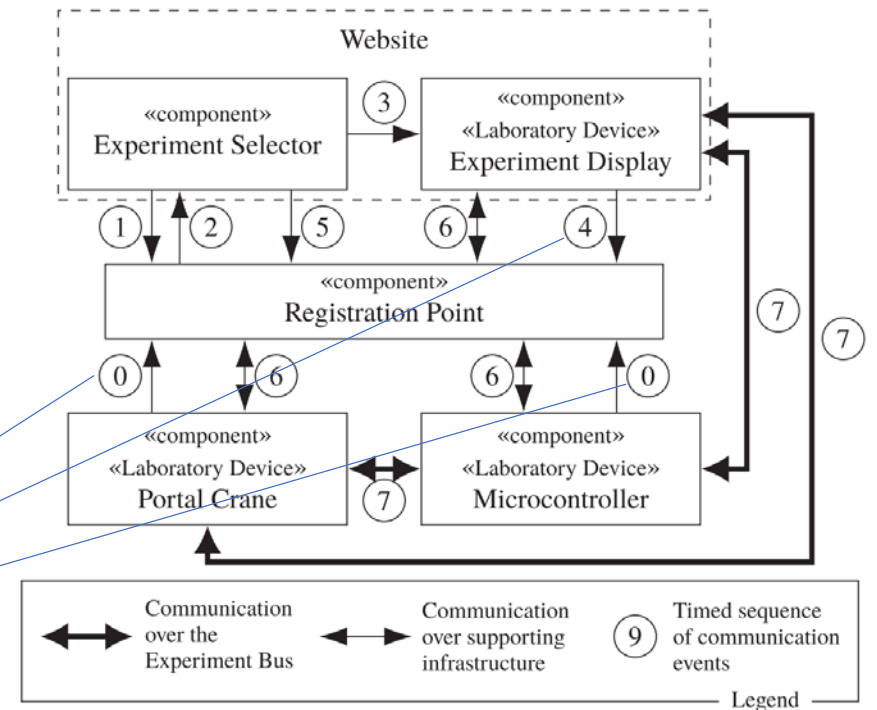
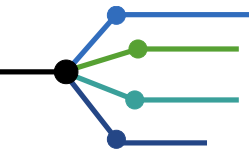
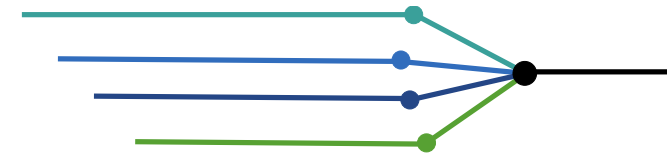


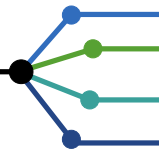
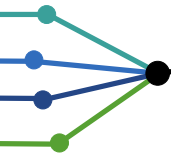
Fig. 10 High level experiment setup



Architekturkonzept



- Laboratory Devices
- **Experiment Konfiguration**
- Kommunikationsaufbau (WebRTC)
- Kommunikation während des Experiments



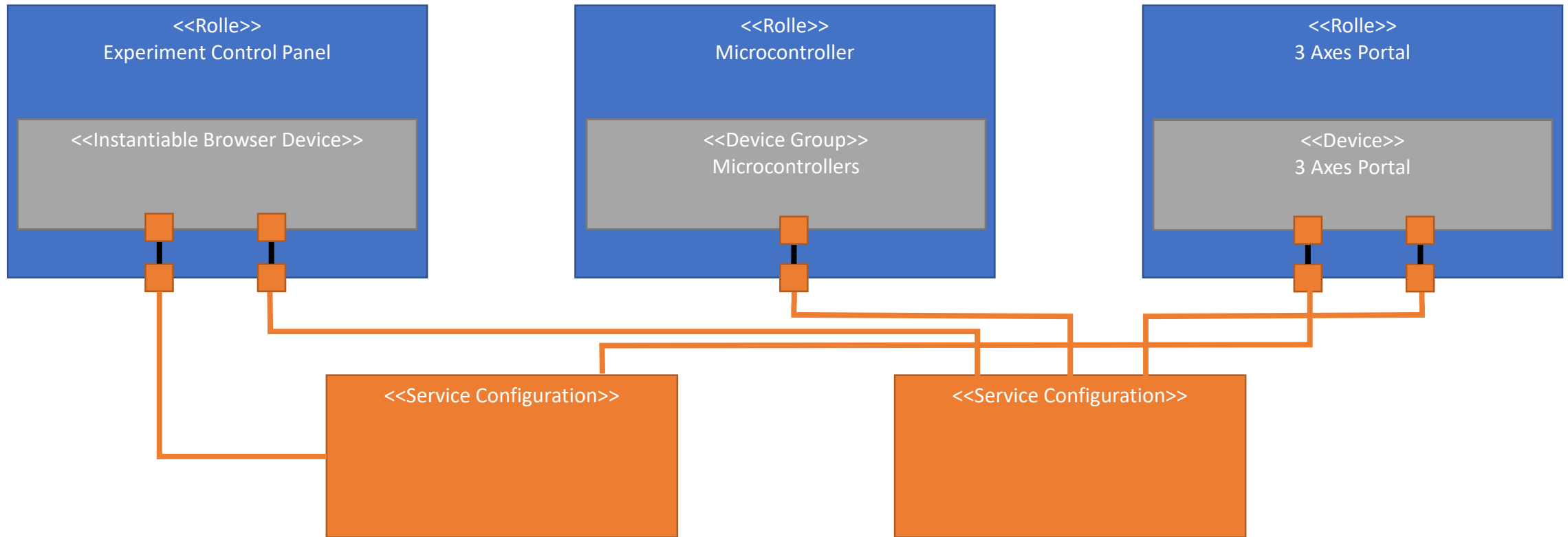
Architekturkonzept

- Experiment Konfiguration



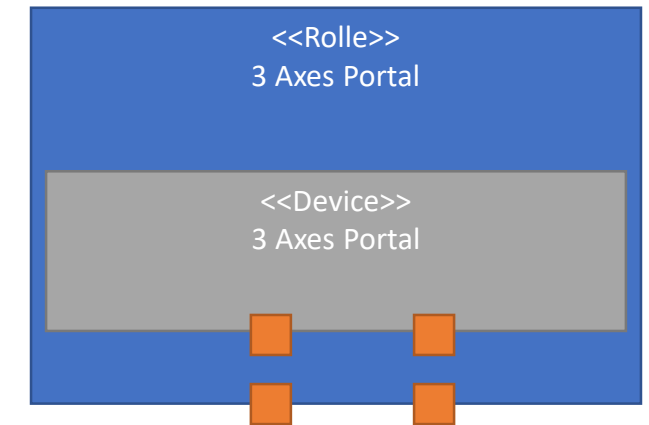
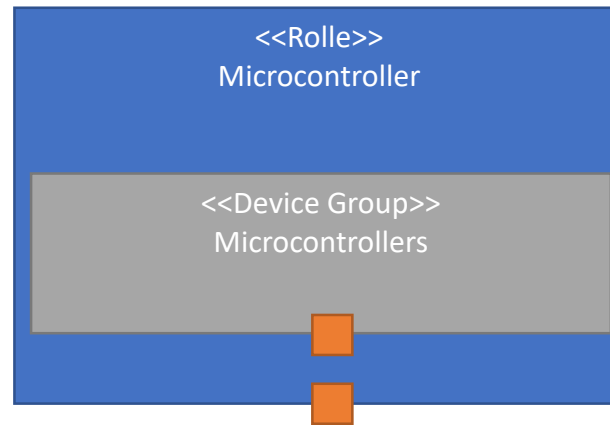
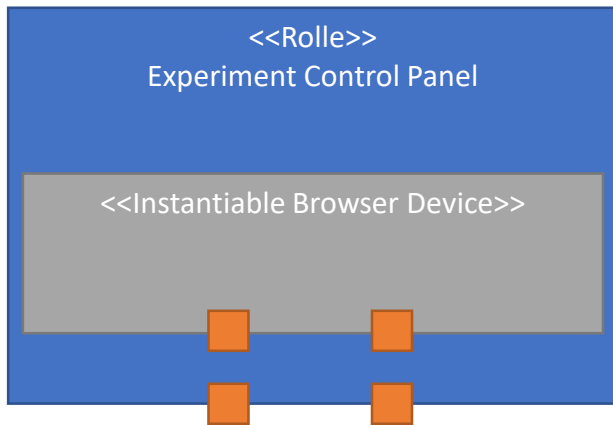
Architekturkonzept

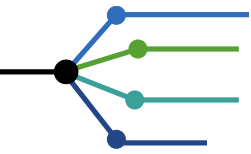
- Experiment Konfiguration



Architekturkonzept

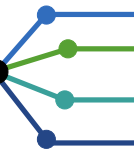
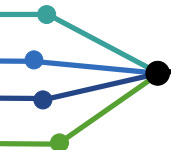
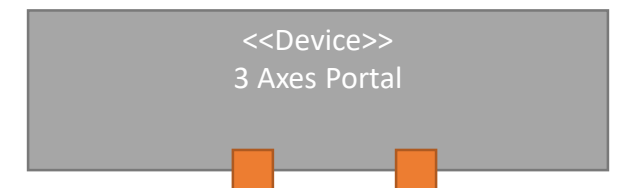
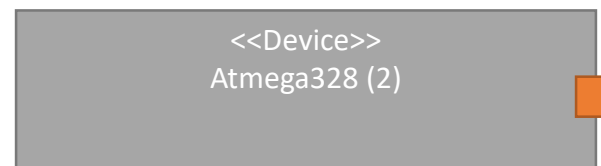
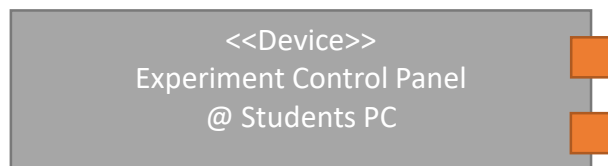
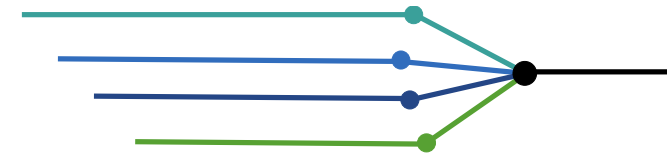
- Experiment Konfiguration





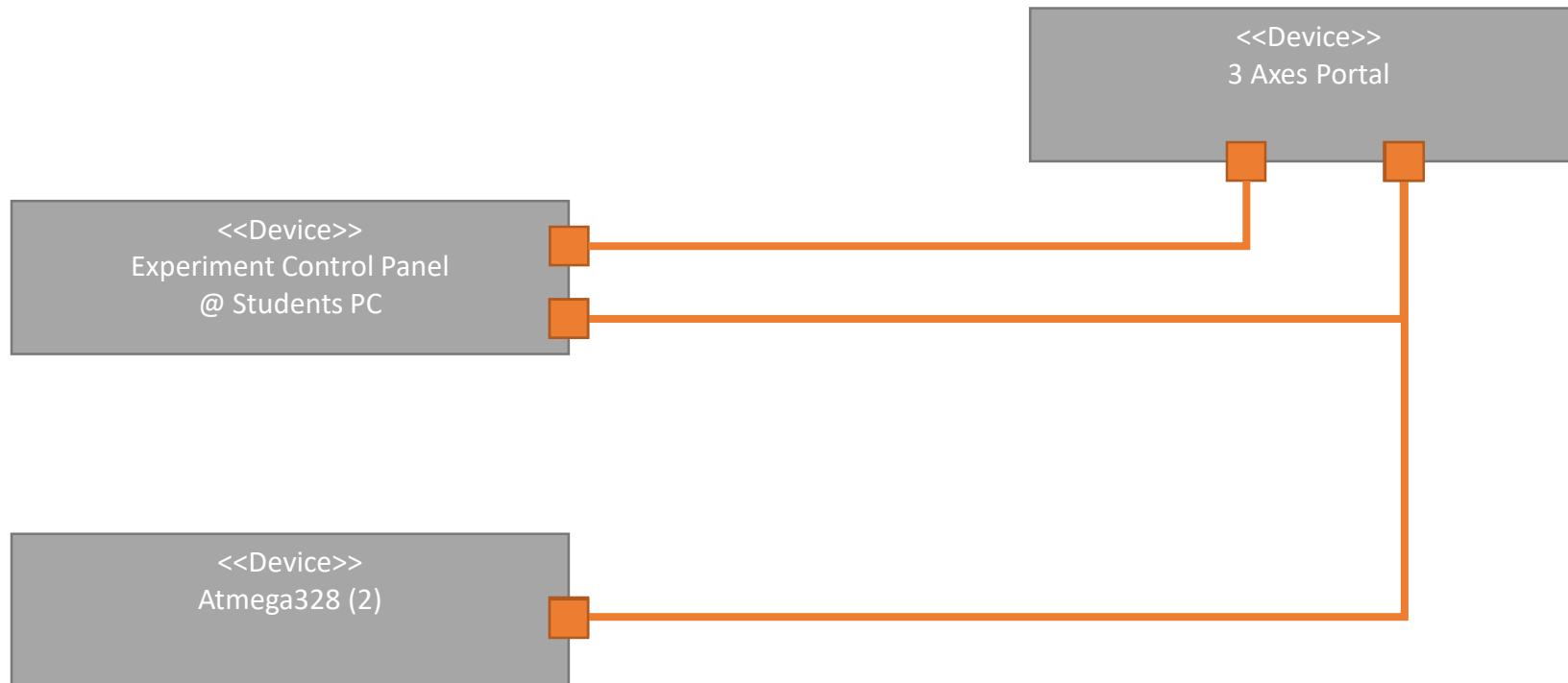
Architekturkonzept

- Experiment Konfiguration



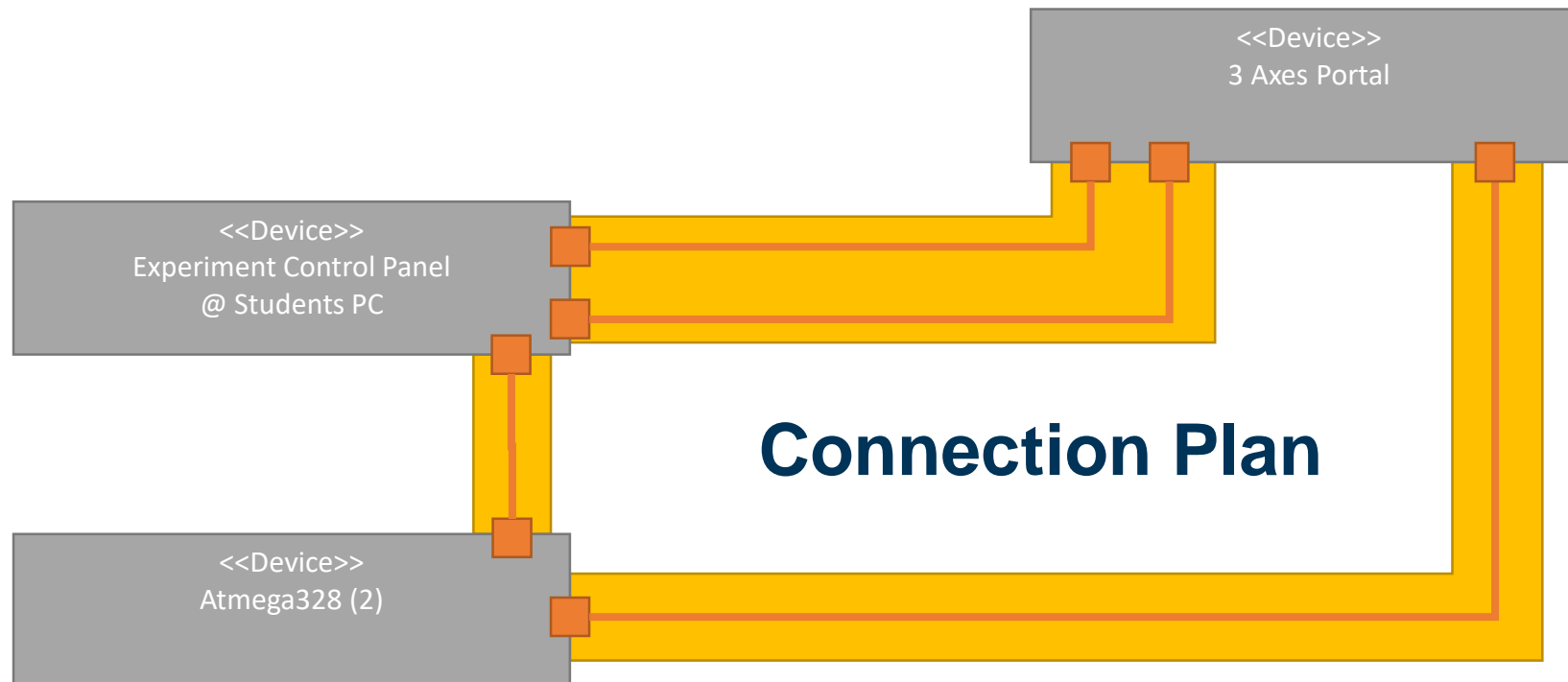
Architekturkonzept

- Experiment Konfiguration



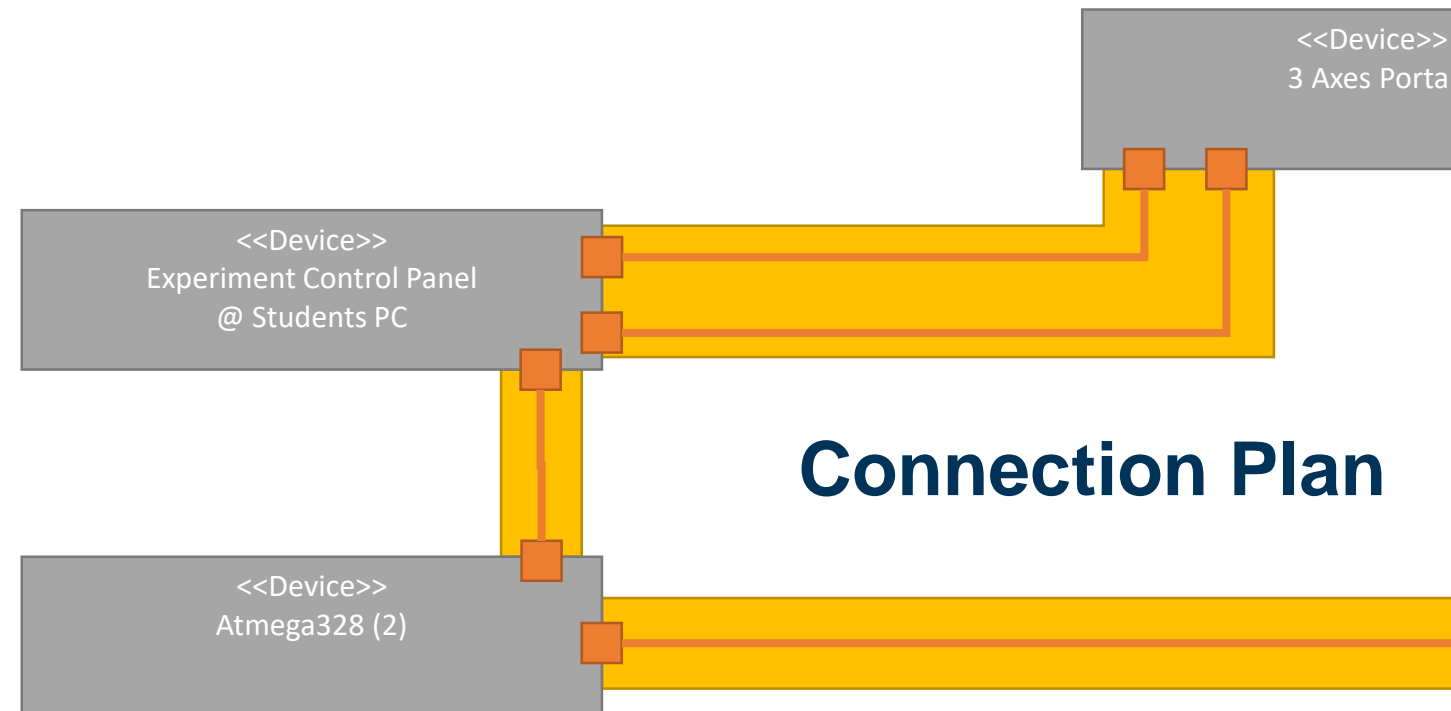
Architekturkonzept

- Experiment Konfiguration



Architekturkonzept

- Experiment Konfiguration
- **Kommunikationsaufbau (WebRTC)**



Connection Plan

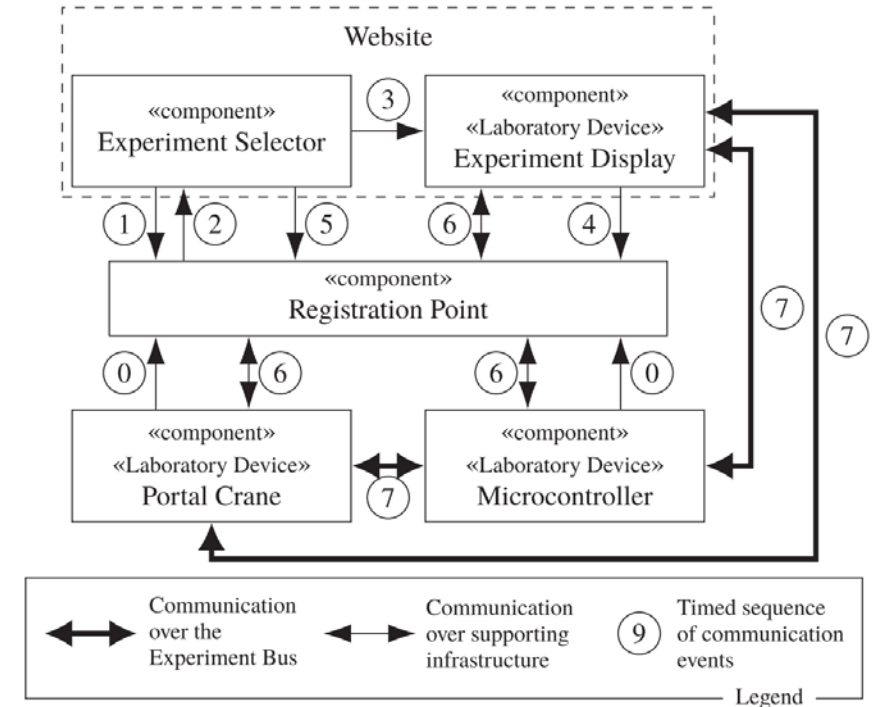


Fig. 10 High level experiment setup



Architekturkonzept

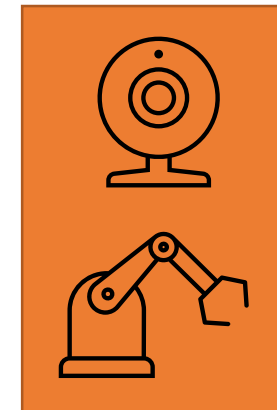
- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- **Kommunikation während des Experiments**



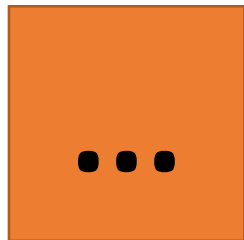
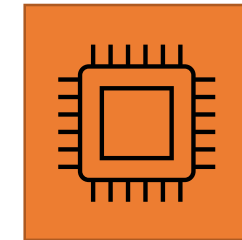
Service Oriented Architektur (SOA)

- jedes Lab-Device
 - bietet Services an oder
 - konsumiert Services

HW-Model



μC, FPGA, SPS User Interface





Architekturkonzept

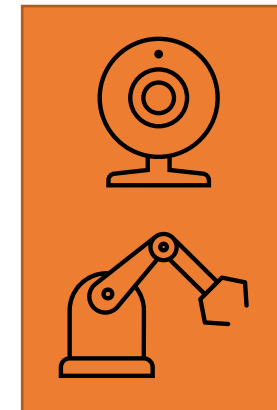
- Laboratory Devices
- Experiment Konfiguration
- Kommunikationsaufbau (WebRTC)
- **Kommunikation während des Experiments**



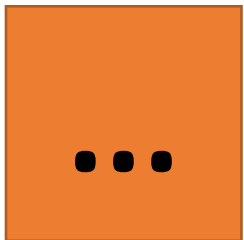
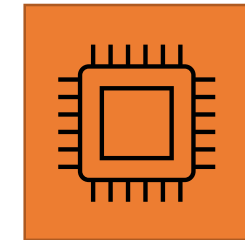
Service Oriented Architektur (SOA)

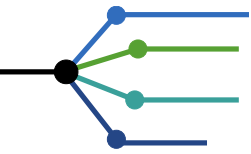
- File-Service
- Electrical-Service
- Webcam-Service

HW-Model

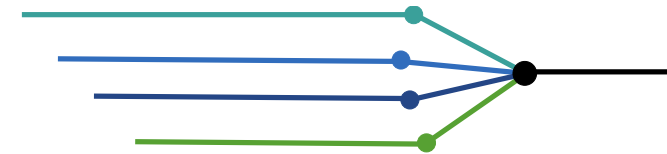


μC, FPGA, SPS User Interface

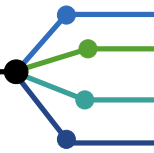
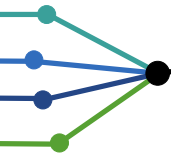




Eigener Laboratory Devices



Beispiel



Eigener Laboratory Devices

```
1 import asyncio
2 from typing import Dict, Optional
3
4 from crosslab.api_client import APIClient
5 from crosslab.soa_client.device_handler import DeviceHandler
6 from crosslab.soa_services.webcam import WebcamService__Producer, GstTrack, UDPTrack
7 from crosslab.soa_services.file import FileService__Consumer, FileServiceEvent
8 from crosslab.soa_services.message import MessageService__Producer
9
10
11 async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29     async def onFile(event: FileServiceEvent):
30         print("Received file of type", event["file_type"])
31         print("File content:", event["content"])
32         await messageService.sendMessage(
33             "Received file of type " + event["file_type"], "error"
34         )
35
36     fileService = FileService__Consumer("file")
37     fileService.on("file", onFile)
38     deviceHandler.add_service(fileService)
39
40     async with APIClient(url) as client:
41         client.set_auth_token(auth_token)
42         deviceHandlerTask = asyncio.create_task(
43             deviceHandler.connect(device_id, client)
44         )
45
46         await deviceHandlerTask
47
48
49 def main():
50     asyncio.run(main_async())
51
52
53 if __name__ == "__main__":
54     main()
55
```



Eigener Laboratory Devices



```
1 import asyncio
2 from typing import Dict, Optional
3
4 from crosslab.api_client import APIClient
5 from crosslab.soa_client.device_handler import DeviceHandler
6 from crosslab.soa_services.webcam import WebcamService__Producer, GstTrack, UDPTrack
7 from crosslab.soa_services.file import FileService__Consumer, FileServiceEvent
8 from crosslab.soa_services.message import MessageService__Producer
9
10
11 async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
```

Eigener Laboratory Devices

```
1 import asyncio
2 from typing import Dict, Optional
3
4 from crosslab.api_client import APIClient
5 from crosslab.soa_client.device_handler import DeviceHandler
6 from crosslab.soa_services.webcam import WebcamService__Producer, GstTrack, UDPTrack
7 from crosslab.soa_services.file import FileService__Consumer, FileServiceEvent
8 from crosslab.soa_services.message import MessageService__Producer
9
10
11 async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
```

Eigener Laboratory Devices

```
1 import asyncio
2 from typing import Dict, Optional
3
4 from crosslab.api_client import APIClient
5 from crosslab.soa_client.device_handler import DeviceHandler
6 from crosslab.soa_services.webcam import WebcamService__Producer, GstTrack, UDPTrack
7 from crosslab.soa_services.file import FileService__Consumer, FileServiceEvent
8 from crosslab.soa_services.message import MessageService__Producer
9
10
11 async def main async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
```

Eigener Laboratory Devices

```
11 - async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29 - async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
```



Eigener Laboratory Devices



```
11 - async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29 - async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
```




Eigener Laboratory Devices



```
11 - async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29 - async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
```



Eigener Laboratory Devices



```
24 deviceHandler.add_service(webcamService)
25
26 messageService = MessageService__Producer("message")
27 deviceHandler.add_service(messageService)
28
29 - async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
36 fileService = FileService__Consumer("file")
37 fileService.on("file", onFile)
38 deviceHandler.add_service(fileService)
39
40 - async with APIClient(url) as client:
41     client.set_auth_token(auth_token)
42     deviceHandlerTask = asyncio.create_task(
43         deviceHandler.connect(device_id, client)
44     )
45
46     await deviceHandlerTask
47
48
```



Eigener Laboratory Devices



```
31     print( File content: , event[ content ])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
36     fileService = FileService__Consumer("file")
37     fileService.on("file", onFile)
38     deviceHandler.add_service(fileService)
39
40     async with APIClient(url) as client:
41         client.set_auth_token(auth_token)
42         deviceHandlerTask = asyncio.create_task(
43             deviceHandler.connect(device_id, client)
44         )
45
46         await deviceHandlerTask
47
48
49     def main():
50         asyncio.run(main_async())
51
52
53     if __name__ == "__main__":
54         main()
55
```



Eigener Laboratory Devices



```
31     print( File content: , event[ content ])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
36     fileService = FileService__Consumer("file")
37     fileService.on("file", onFile)
38     deviceHandler.add_service(fileService)
39
40     async with APIClient(url) as client:
41         client.set_auth_token(auth_token)
42         deviceHandlerTask = asyncio.create_task(
43             deviceHandler.connect(device_id, client)
44         )
45
46     await deviceHandlerTask
47
48
49     def main():
50         asyncio.run(main_async())
51
52
53     if __name__ == "__main__":
54         main()
55
```

Eigener Laboratory Devices

```
1 import asyncio
2 from typing import Dict, Optional
3
4 from crosslab.api_client import APIClient
5 from crosslab.soa_client.device_handler import DeviceHandler
6 from crosslab.soa_services.webcam import WebcamService__Producer, GstTrack, UDPTrack
7 from crosslab.soa_services.file import FileService__Consumer, FileServiceEvent
8 from crosslab.soa_services.message import MessageService__Producer
9
10
11 async def main_async():
12     auth_token = "0235bbfd-01b8-4981-96f2-fcef3f95bb2d"
13     device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"
14     url = "https://api.goldi-labs.de"
15
16     deviceHandler = DeviceHandler()
17
18     webcamService = WebcamService__Producer(
19         GstTrack(
20             "videotestsrc is-live=true pattern=ball ! videoconvert ! queue ! x264enc tune=zerolatency ! 'video/x-h264,level=(string)4'"
21         ),
22         "webcam",
23     )
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29     async def onFile(event: FileServiceEvent):
30         print("Received file of type", event["file_type"])
31         print("File content:", event["content"])
32         await messageService.sendMessage(
33             "Received file of type " + event["file_type"], "error"
34         )
35
36     fileService = FileService__Consumer("file")
37     fileService.on("file", onFile)
38     deviceHandler.add_service(fileService)
39
40     async with APIClient(url) as client:
41         client.set_auth_token(auth_token)
42         deviceHandlerTask = asyncio.create_task(
43             deviceHandler.connect(device_id, client)
44         )
45
46         await deviceHandlerTask
47
48
49 def main():
50     asyncio.run(main_async())
51
52
53 if __name__ == "__main__":
54     main()
55
```

Experiment Konfiguration

```
1 |  
2 "status": "running",  
3 "roles": [  
4   {  
5     "name": "NAK Robot",  
6     "template_device": "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7dee693"  
7   },  
8   {  
9     "name": "ECP",  
10    "template_device": "https://api.goldi-labs.de/devices/91c48ca7-d666-4f9b-9d3a-628f09daa058"  
11  }  
12 ],  
13 "serviceConfigurations": [  
14   {  
15     "serviceType": "https://api.goldi-labs.de/serviceTypes/file",  
16     "configuration": {},  
17     "participants": [  
18       {  
19         "serviceId": "file",  
20         "role": "ECP",  
21         "config": {}  
22       },  
23       {  
24         "serviceId": "file",  
25         "role": "NAK Robot",  
26         "config": {}  
27       }  
28     ],  
29     "id": "3182f339-4968-4413-a0f0-cbf559deca74"  
30   },  
31   {  
32     "serviceType": "https://api.goldi-labs.de/serviceTypes/webcam",  
33     "configuration": {},  
34     "participants": [  
35       {  
36         "serviceId": "webcam",  
37         "role": "ECP",  
38         "config": {}  
39       },  
40       {  
41         "serviceId": "webcam",  
42         "role": "NAK Robot",  
43         "config": {}  
44       }  
45     ],  
46     "id": "ce2ca59d-2c25-4e52-bd03-829f9ab6fa10"  
47   },  
48   {  
49     "serviceType": "https://api.goldi-labs.de/serviceTypes/message",  
50     "configuration": {},  
51     "participants": [  
52       {  
53         "serviceId": "message",  
54         "role": "ECP",  
55         "config": {}  
56       },  
57       {  
58         "serviceId": "message",  
59         "role": "NAK Robot",  
60         "config": {}  
61       }  
62     ],  
63     "id": "ce2ca59d-2c25-4e52-bd03-829f9ab6fa91"  
64   }  
65 ]  
66 |
```



Experiment Konfiguration



```
1 {  
2   "status": "running",  
3   "roles": [  
4     {  
5       "name": "NAK Robot",  
6       "template_device": "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"  
7     },  
8     {  
9       "name": "ECP",  
10      "template_device": "https://api.goldi-labs.de/devices/91c48ca7-d666-4f9b-9d3a-628f09daa058"  
11    }  
12  ],  
13  "serviceConfigurations": [  
14    {  
15      "serviceType": "https://api.goldi-labs.de/serviceTypes/file",  
16      "configuration": {},  
17      "participants": [  
18        {  
19          "serviceId": "file",  
20          "role": "ECP",  
21          "config": {}  
22        },  
23        {  
24          "serviceId": "file",
```

Experiment Konfiguration

```
1 {  
2   "status": "running",  
3   "roles": [  
4     {  
5       "name": "NAK Robot",  
6       "template_device": "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b-7a0ee7d6e693"  
7     },  
8     {  
9       "name": "ECP",  
10      "template_device": "https://api.goldi-labs.de/devices/91c48ca7-d666-4f9b-9d3a-628f09daa058"  
11    }  
12  ],  
13  "serviceConfigurations": [  
14    {  
15      "serviceType": "https://api.goldi-labs.de/serviceTypes/file",  
16      "configuration": {},  
17      "participants": [  
18        {  
19          "serviceId": "file",  
20          "role": "ECP",  
21          "config": {}  
22        },  
23        {  
24          "serviceId": "file",
```


Experiment Konfiguration

```
9   "name": "ECP",
10  "template_device": "https://api.goldi-labs.de/devices/91c48ca7-d666-4f9b-9d3a-628f09daa058"
11  }
12  },
13  "serviceConfigurations": [
14    {
15      "serviceType": "https://api.goldi-labs.de/serviceTypes/file",
16      "configuration": {},
17      "participants": [
18        {
19          "serviceId": "file",
20          "role": "ECP",
21          "config": {}
22        },
23        {
24          "serviceId": "file",
25          "role": "NAK Robot",
26          "config": {}
27        }
28      ],
29      "id": "3182f339-4968-4413-a0f0-cbf559deca74"
30    },
31    {
32      "serviceType": "https://api.goldi-labs.de/serviceTypes/webcam",
```

Experiment Konfiguration

```
9     "name": "ECP",
10     "template_device": "https://api.goldi-labs.de/devices/91c48ca7-d666-4d-80-30-1b3000000000",
11 }
12 },
13 "serviceConfigurations": [
14 {
15     "serviceType": "https://api.goldi-labs.de/serviceTypes/file",
16     "configuration": {},
17     "participants": [
18 {
19     "serviceId": "file",
20     "role": "ECP",
21     "config": {}
22 },
23 {
24     "serviceId": "file",
25     "role": "NAK Robot",
26     "config": {}
27 },
28 ],
29 "id": "3182f339-4968-4413-a0f0-cbf559deca74"
30 },
31 {
32     "serviceType": "https://api.goldi-labs.de/serviceTypes/webcam",
```

```
24     deviceHandler.add_service(webcamService)
25
26     messageService = MessageService__Producer("message")
27     deviceHandler.add_service(messageService)
28
29     async def onFile(event: FileServiceEvent):
30         print("Received file of type", event["file_type"])
31         print("File content:", event["content"])
32         await messageService.sendMessage(
33             "Received file of type " + event["file_type"], "error"
34         )
35
36     fileService = FileService__Consumer("file")
37     fileService.on("file", onFile)
38     deviceHandler.add_service(fileService)
39
40     async with APIClient(url) as client:
41         client.set_auth_token(auth_token)
42         deviceHandlerTask = asyncio.create_task(
43             deviceHandler.connect(device_id, client)
44         )
45
46         await deviceHandlerTask
47
48
```

Experiment Konfiguration

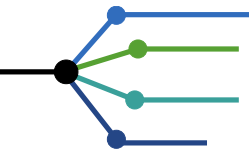
```
28     },
29     "id": "3182f339-4968-4413-a0f0-cbf559deca74"
30 }
31 {
32     "serviceType": "https://api.goldi-labs.de/serviceTypes/webcam",
33     "configuration": {},
34     "participants": [
35         {
36             "serviceId": "webcam",
37             "role": "ECP",
38             "config": {}
39         },
40         {
41             "serviceId": "webcam",
42             "role": "NAK Robot",
43             "config": {}
44         }
45     ],
46     "id": "ce2ca59d-2c25-4e52-bd03-829f9ab6fa10"
47 },
48 {
49     "serviceType": "https://api.goldi-labs.de/serviceTypes/message",
50     "configuration": {},
51     "participants": [
```

```
13 device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b
14 url = "https://api.goldi-labs.de"
15
16 deviceHandler = DeviceHandler()
17
18 webcamService = WebcamService__Producer(
19     GstTrack(
20         "videotestsrc is-live=true pattern=ball ! videoconvert ! queue
21     ),
22     "webcam",
23 )
24 deviceHandler.add_service(webcamService)
25
26 messageService = MessageService__Producer("message")
27 deviceHandler.add_service(messageService)
28
29 async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
36 fileService = FileService__Consumer("file")
37 fileService.on("file", onFile)
```

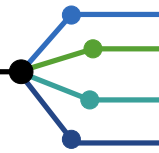
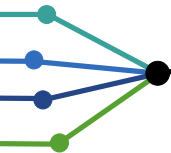
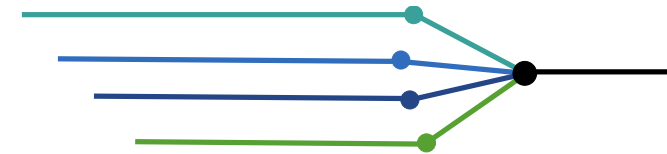
Experiment Konfiguration

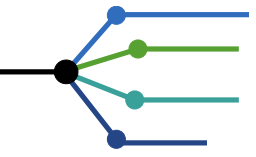
```
42     "role": "NAK Robot",
43     "config": {}
44   },
45 ],
46 "id": "ce2ca59d-2c25-4e52-bd03-829f9ab6fa10"
47 }
48 {
49   "serviceType": "https://api.goldi-labs.de/serviceTypes/message",
50   "configuration": {},
51   "participants": [
52     {
53       "serviceId": "message",
54       "role": "ECP",
55       "config": {}
56     },
57     {
58       "serviceId": "message",
59       "role": "NAK Robot",
60       "config": {}
61     }
62   ],
63   "id": "ce2ca59d-2c25-4e52-bd03-829f9ab6fa91"
64 }
65 }
```

```
13 device_id = "https://api.goldi-labs.de/devices/3eacc285-43d2-402e-bc5b
14 url = "https://api.goldi-labs.de"
15
16 deviceHandler = DeviceHandler()
17
18 webcamService = WebcamService__Producer(
19     GstTrack(
20         "videotestsrc is-live=true pattern=ball ! videoconvert ! queue
21     ),
22     "webcam",
23 )
24 deviceHandler.add_service(webcamService)
25
26 messageService = MessageService__Producer("message")
27 deviceHandler.add_service(messageService)
28
29 async def onFile(event: FileServiceEvent):
30     print("Received file of type", event["file_type"])
31     print("File content:", event["content"])
32     await messageService.sendMessage(
33         "Received file of type " + event["file_type"], "error"
34     )
35
36 fileService = FileService__Consumer("file")
37 fileService.on("file", onFile)
```

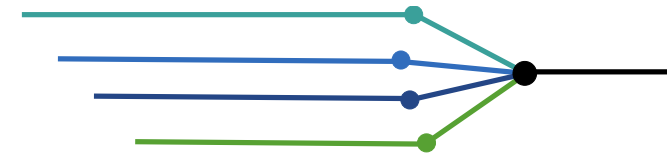


Nutzungskonzept: GOLDi 2.0

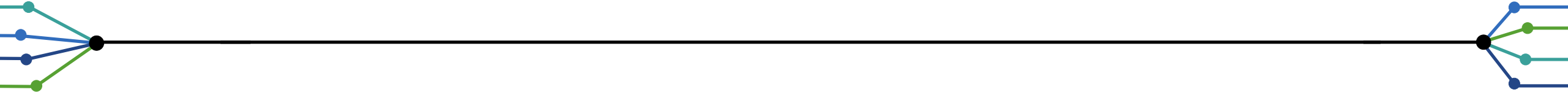


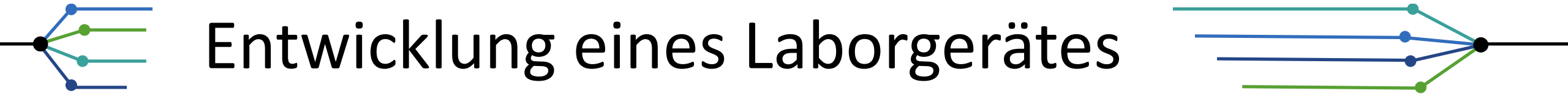


Entwicklung eines Laborgerätes



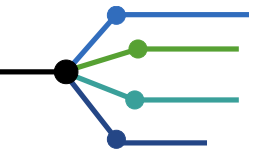
- Beispiel: Ampelsteuerung



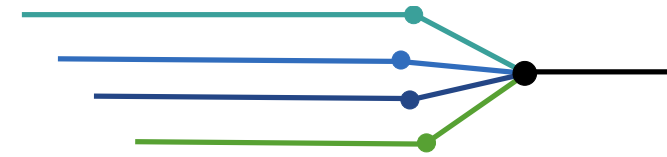


- Beispiel: Ampelsteuerung

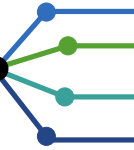
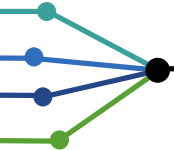




Entwicklung eines Laborgerätes



- Voraussetzung:
 - Python 3.8-3.10
 - Für Webcam: GStreamer



Python

PSF

Docs

PyPI

Jobs

Community



Donate

GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

Download the latest version for Windows

Download Python 3.11.3

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

Want to help test development versions of Python? [Prereleases](#), [Docker images](#)



Active Python Releases

For more information visit the [Python Developer's Guide](#).

3.8	security	2019-10-14	2024-10	PEP 569
3.7	security	2018-06-27	2023-06-27	PEP 537

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
Python 3.10.11	April 5, 2023	Download	Release Notes
Python 3.11.3	April 5, 2023	Download	Release Notes
Python 3.10.10	Feb. 8, 2023	Download	Release Notes
Python 3.11.2	Feb. 8, 2023	Download	Release Notes
Python 3.11.1	Dec. 6, 2022	Download	Release Notes
Python 3.10.9	Dec. 6, 2022	Download	Release Notes
Python 3.9.16	Dec. 6, 2022	Download	Release Notes

[View older releases](#)

[Full Changelog](#)

Files



Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	
Gzipped source tarball	Source release		6dbe644dd1a520d9853cf6648084c346	26071329	SIG	CRT	SIG
XZ compressed source tarball	Source release		7bf85df71bbe7f95e5370b983e6ae684	19627028	SIG	CRT	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	892634724ab799569b512082c8f48c83	41005648	SIG	CRT	SIG
Windows embeddable package (32-bit)	Windows		a681a7f9b242fe35b4d96d79e15e57d6	7663448	SIG	CRT	SIG
Windows embeddable package (64-bit)	Windows		f38a9e7e02a992daa62569b758d0a388	8625602	SIG	CRT	SIG
Windows help file	Windows		448f8401ade49a7e2156d02512f2f9bf	9391521	SIG	CRT	SIG
Windows installer (32-bit)	Windows		a81b81687bc2575c05a30f4b31d6ea00	27859200	SIG	CRT	SIG
Windows installer (64-bit)	Windows	Recommended	9735797853cba809b13c8396c91354a0	29010904	SIG	CRT	SIG

[Full Changelog](#)

Files


- Version
- [Gzipped source tarball](#)
 - [XZ compressed source tarball](#)
 - [macOS 64-bit universal2 installer](#)
 - [Windows embeddable package \(32-bit\)](#)
 - [Windows embeddable package \(64-bit\)](#)
 - [Windows help file](#)
 - [Windows installer \(32-bit\)](#)
 - [Windows installer \(64-bit\)](#)

Python 3.10.10 (64-bit) Setup



Install Python 3.10.10 (64-bit)

Select **Install Now** to install Python with default settings, or choose **Customize** to enable or disable features.

 **Install Now**
C:\Users\jona3814_admin\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ **Customize installation**
Choose location and features

☒ Use admin privileges when installing py.exe
☐ Add python.exe to PATH


Cancel

File Size	GPG	Sigstore	
26071329	SIG	CRT	SIG
19627028	SIG	CRT	SIG
41005648	SIG	CRT	SIG
7663448	SIG	CRT	SIG
8625602	SIG	CRT	SIG
9391521	SIG	CRT	SIG
27859200	SIG	CRT	SIG
29010904	SIG	CRT	SIG

[Full Changelog](#)

Files


- Version
- [Gzipped source tarball](#)
 - [XZ compressed source tarball](#)
 - [macOS 64-bit universal2 installer](#)
 - [Windows embeddable package \(32-bit\)](#)
 - [Windows embeddable package \(64-bit\)](#)
 - [Windows help file](#)
 - [Windows installer \(32-bit\)](#)
 - [Windows installer \(64-bit\)](#)




python
for
windows

Install Python 3.10.10 (64-bit)

Select **Install Now** to install Python with default settings, or choose **Customize** to enable or disable features.

 **Install Now**
C:\Users\jona3814_admin\AppData\Local\Programs\Python\Python310

Includes IDLE, pip and documentation
Creates shortcuts and file associations

 **Customize installation**
Choose location and features

☒ Use admin privileges when installing py.exe
☒ Add python.exe to PATH

Cancel



File Size	GPG	Sigstore	
26071329	SIG	CRT	SIG
19627028	SIG	CRT	SIG
41005648	SIG	CRT	SIG
7663448	SIG	CRT	SIG
8625602	SIG	CRT	SIG
9391521	SIG	CRT	SIG
27859200	SIG	CRT	SIG
29010904	SIG	CRT	SIG

[Full Changelog](#)

Files

- Version
- [Gzipped source tarball](#)
 - [XZ compressed source tarball](#)
 - [macOS 64-bit universal2 installer](#)
 - [Windows embeddable package \(32-bit\)](#)
 - [Windows embeddable package \(64-bit\)](#)
 - [Windows help file](#)
 - [Windows installer \(32-bit\)](#)
 - [Windows installer \(64-bit\)](#)


Python 3.10.10 (64-bit) Setup



Setup was successful

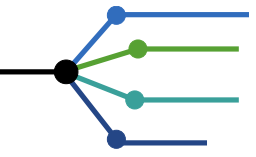
New to Python? Start with the [online tutorial](#) and [documentation](#). At your terminal, type "py" to launch Python, or search for Python in your Start menu.

See [what's new](#) in this release, or find more info about [using Python on Windows](#).

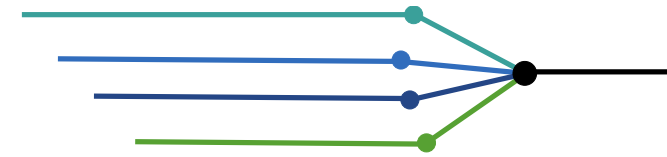
 **Disable path length limit**
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Close

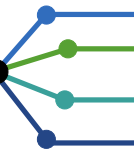
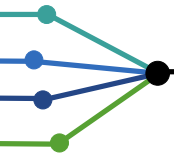
File Size	GPG	Sigstore	
26071329	SIG	CRT	SIG
19627028	SIG	CRT	SIG
41005648	SIG	CRT	SIG
7663448	SIG	CRT	SIG
8625602	SIG	CRT	SIG
9391521	SIG	CRT	SIG
27859200	SIG	CRT	SIG
29010904	SIG	CRT	SIG

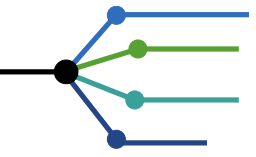


Entwicklung eines Laborgerätes

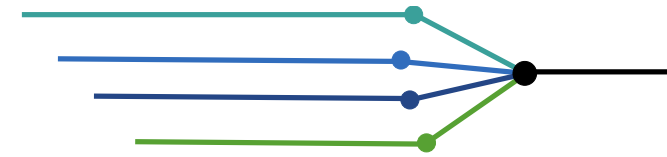


- Voraussetzung:
 - Python 3.8-3.10
 - Für Webcam: Gstreamer → Nur für Webcam Installation später

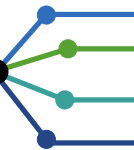
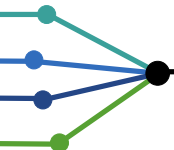




Entwicklung eines Laborgerätes



- Voraussetzung:
 - Python 3.8-3.10
 - Für Webcam: GStreamer → Nur für Webcam Installation später
- Entwicklungsumgebung
 - Hier: VS Code



Version 1.78 is now available! Read about the new features and fixes from April.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build


Web, Insiders edition, or other platforms


By using VS Code, you agree to its
license and privacy statement.


File Edit Selection View Go Debug Terminal Help serviceWorker.js - create-react-app - Visual Studio Code - In...


EXTENSIONS: MARKETPLACE


@sort:installs


**Python** 2019.6.24221 54.9M 4.5
Linting, Debugging (multi-threaded, ...
Microsoft Install


**GitLens — Git sup...** 9.8.5 23.1M 5
Supercharge the Git capabilities buil...
Eric Amodio Install


**C/C++** 0.24.0 23M 3.5
C/C++ IntelliSense, debugging, and ...
Microsoft Install

**ESLint** 1.9.0 21.9M 4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer Install

**Debugger for Ch...** 4.11.6 20.6M 4
Debug your JavaScript code in the C...
Microsoft Install

**Language Supp...** 0.47.0 18.7M 4.5
Java Linting, Intellisense, formatting, ...
Red Hat Install

**vscode-icons** 8.8.0 17.2M 5
Icons for Visual Studio Code
VSCode Icons Team Install

**Vetur** 0.21.1 17M 4.5
Vue tooling for VS Code

JS App.js JS index.js JS serviceWorker.js

src > JS serviceWorker.js > register > window.addEventListener('load') callback

39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

checkValidServiceWorker(swUrl, config);

// Add some additional logging to localhost, p
// service worker/PWA documentation.

navigator.serviceWorker.ready.then(() => {
 product
 productSub
 removeSiteSpecificTrackingException
 removeWebWideTrackingException
 requestMediaKeySystemAccess
 sendBeacon
 serviceWorker (property) Navigator.serviceWorke...
 storage
 storeSiteSpecificTrackingException
 storeWebWideTrackingException
 userAgent
 vendor
})


function registerValidSW(swUrl, config) {
 navigator.serviceWorker
 .register(swUrl)
 .then(registration => {


TERMINAL ... 1: node


You can now view create-react-app in the browser.


EXTENSIONS: MARKETPLACE


python


**Python**
IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code f...
Microsoft Install


**Python Indent**
Correct Python indentation
Kevin Rose Install


**Python Extens...**
Popular Visual Studio Code ...
Don Jayamanne Install

**Python for VS...**
Python language extension ...
Thomas Haakon... Install


**autoDocstring...**
Generates python docstring...
Nils Werner Install

**Python Enviro...**
View and manage Python e...
Don Jayamanne Install

**Python**
Extensions for Python
shiro Install









**Python Previ...**
Provide Preview for Python ...

Extension: Python




Python

v2023.8.0

Microsoft  microsoft.com |  86,398,338 |       (542)

IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code f...

Install 

DETAILS

FEATURE CONTRIBUTIONS

CHANGELOG

EXTENSION PACK

Python extension for Visual Studio Code

A [Visual Studio Code extension](#) with rich support for the [Python language](#) (for all [actively supported versions](#) of the language: ≥ 3.7), including features such as IntelliSense (Pylance), linting, debugging, code navigation, code formatting, refactoring, variable explorer, test explorer, and more!

Support for vscode.dev

The Python extension does offer [some support](#) when running on vscode.dev (which includes github.dev). This includes partial IntelliSense for open files in the editor.

Installed extensions

The Python extension will automatically install the [Pylance](#) and [Jupyter](#) extensions to give you the best experience when working with Python files and Jupyter notebooks. However, Pylance is an optional dependency, meaning the Python extension will remain fully functional

Categories

Programming Languages

Linters

Debuggers

Formatters

Data Science

Machine Learning

Notebooks

Extension Resources

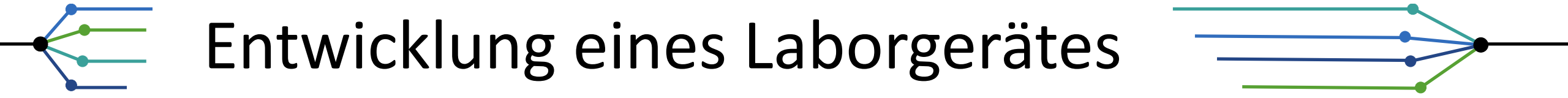
Marketplace

Repository

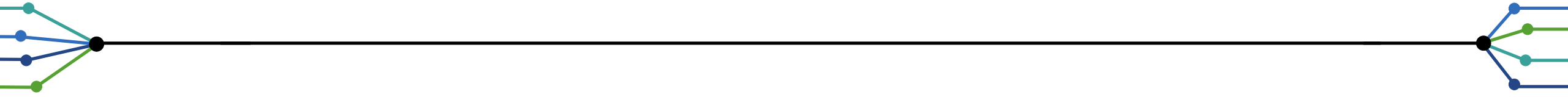
License

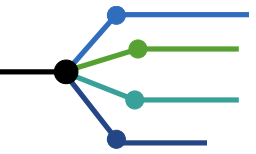
Microsoft

More Info

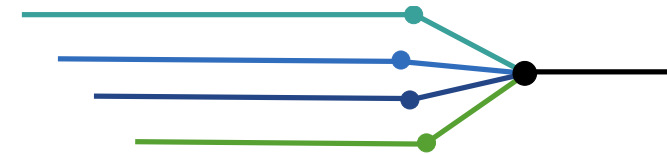


Live

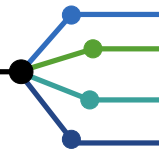
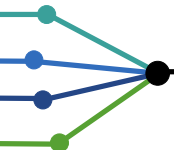




Entwicklung eines Laborgerätes



- Voraussetzung:
 - Python 3.8-3.10
 - Für Webcam: GStreamer
- Entwicklungsumgebung
 - Hier: VS Code





- Home
- Features
- News
- Annual Conference
- Planet (Blogs)
- Download
- Applications

- Security Center
- GitLab
- Developers
- Documentation
- Mailing Lists
- File a Bug
- Bug Lists

- Artwork
- @gstreamer on Twitter
- @gstreamer on Mastodon

Download GStreamer

If you're on Linux or a BSD variant, you can install GStreamer using your package manager.

For other platforms, specifically [Windows](#), [macOS](#), [Android](#), and [iOS](#), we provide binary releases in the form of official installers or tarballs maintained by the GStreamer project.

Windows

Binary releases in the form of MSI installers are available. The installers are split into runtime and development packages. For development, you will want to install both packages.

- MSVC 64-bit (VS 2019, Release CRT)
 - [1.22.3 runtime installer](#)
 - [1.22.3 development installer](#)
- MSVC 32-bit (VS 2019, Release CRT)
 - [1.22.3 runtime installer](#)
 - [1.22.3 development installer](#)
- MinGW 64-bit
 - [1.22.3 runtime installer](#)
 - [1.22.3 development installer](#)
- MinGW 32-bit
 - [1.22.3 runtime installer](#)
 - [1.22.3 development installer](#)

For each of the above listed targets, a [zip file with .msm modules](#) is available for integration into your own WiX-based app installer.

If you are not sure which to pick between MSVC and MinGW, just pick MSVC. However, do see the [toolchain compatibility notes](#) below which may affect you based on what toolchain your app will be built with.

NOTE: The library names in MSVC are different from MinGW; specifically the DLLs are of the form `foo.dll` instead of `libfoo.dll`.

NOTE: [GstSharp .NET bindings](#) require the MSVC binaries starting with 1.18.

NOTE: Some of the plugins shipped with the MSVC binaries link to non-gstreamer libraries built with MinGW because they are built with Autotools. [See below](#) for what this means for your application.

[Older 1.x binary releases](#) are also available.



Alle Apps Dokumente Web Mehr ▾

Höchste Übereinstimmung

Systemumgebungsvariablen bearbeiten

Systemsteuerung

Einstellungen

Umgebungsvariablen für dieses Konto bearbeiten

Einstellungen für die Umgebungsfreigabe

Einstellungen für die gemeinsame Nutzung

Web durchsuchen

umgebung - Webergebnisse anzeigen

umgebungsvariablen

umgebung englisch

umgebung synonym

umgebungsfreigabe

umgebungsdruck

umgebungsvariablen bearbeiten

Systemumgebungsvariablen bearbeiten

Systemsteuerung

Öffnen

the form of official installers or tarballs maintained by the GStreamer project.

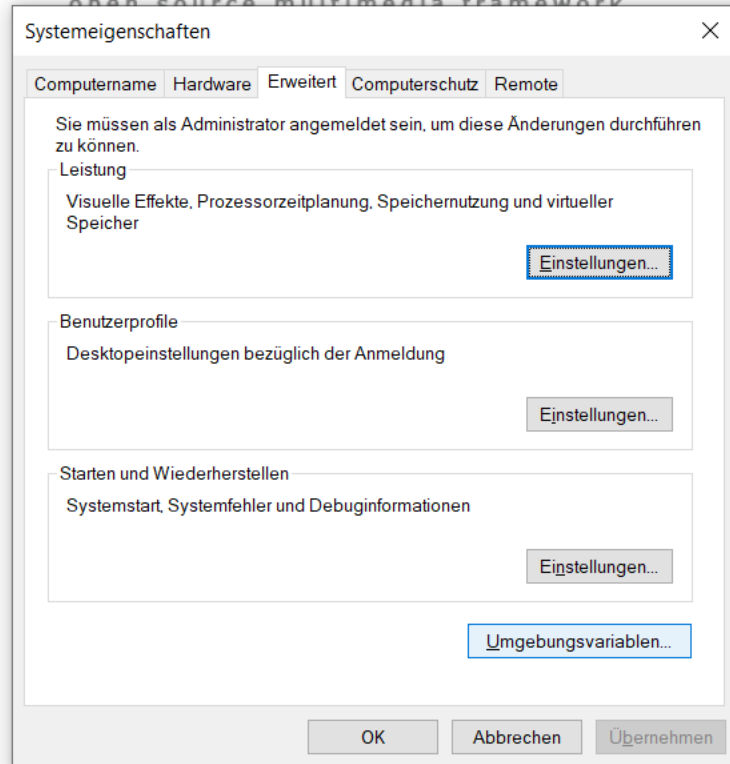
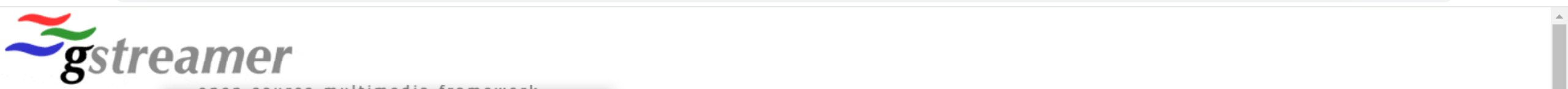
development packages. For development, you will want to install both packages.

own WiX-based app installer.

[toolchain compatibility notes](#) below which may affect you based on what toolchain your app will be

foo.dll instead of libfoo.dll.

MinGW because they are built with Autotools. [See below](#) for what this means for your application.



using your package manager.

On iOS, we provide binary releases in the form of official installers or tarballs maintained by the GStreamer project.

Installers are split into runtime and development packages. For development, you will want to install both packages.

For each of the above listed targets, [a zip file with .msm modules](#) is available for integration into your own WiX-based app installer.

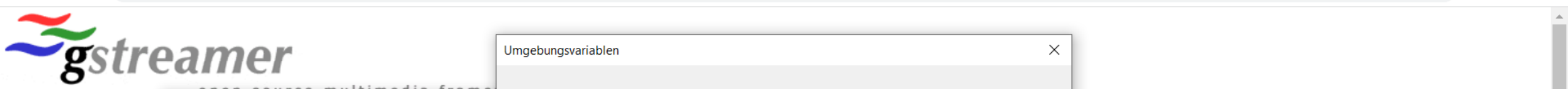
If you are not sure which to pick between MSVC and MinGW, just pick MSVC. However, do see the [toolchain compatibility notes](#) below which may affect you based on what toolchain your app will be built with.

NOTE: The library names in MSVC are different from MinGW; specifically the DLLs are of the form `foo.dll` instead of `libfoo.dll`.

NOTE: [GstSharp .NET bindings](#) require the MSVC binaries starting with 1.18.

NOTE: Some of the plugins shipped with the MSVC binaries link to non-gstreamer libraries built with MinGW because they are built with Autotools. [See below](#) for what this means for your application.

[Older 1.x binary releases](#) are also available.



Home

Features

News

Annual Conference

Planet (Blogs)

Download

Applications

Security Center

GitLab

Developers

Documentation

Mailing Lists

File a Bug

Bug Lists

Artwork

@gstreamer on Twitter

@gstreamer on Mastodon

Systemeigenschaften

Computernamen Hardware Erweitert Computerschutz Remote

Sie müssen als Administrator angemeldet sein, um diese Änderungen zu können.

Leistung

Visuelle Effekte, Prozessorzeitplanung, Speichernutzung und v Speicher

Benutzerprofile

Desktopeinstellungen bezüglich der Anmeldung

Starten und Wiederherstellen

Systemstart, Systemfehler und Debuginformationen

Umgebungsvariablen

OK Abbrechen

Umgebungsvariablen

Benutzervariablen für jona3814_admin

Variable	Wert
ChocolateyLastPathUpdate	133293230856544579
OneDrive	C:\Users\jona3814_admin\OneDrive
OPENSSL_CONF	C:\Program Files\OpenSSL-Win64\bin\openssl.cfg
Path	C:\Users\jona3814_admin\AppData\Local\Programs\Python\P...
TEMP	C:\Users\jona3814_admin\AppData\Local\Temp
TMP	C:\Users\jona3814_admin\AppData\Local\Temp

Neu... Bearbeiten... Löschen

Systemvariablen

Variable	Wert
NUMBER_OF_PROCESSORS	16
OS	Windows_NT
Path	C:\ProgramData\Anaconda3;C:\ProgramData\Anaconda3\Lib...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTU...	AMD64
PROCESSOR_IDENTIFIER	AMD64 Family 25 Model 80 Stepping 0, AuthenticAMD
PROCESSOR_LEVEL	25
PROCESSOR_REVISION	5000

Neu... Bearbeiten... Löschen

OK Abbrechen

For each of the above listed targets, [a zip file](#) will be provided. These files are built with the MSVC toolchain, which may affect you based on what toolchain your app will be built with.

NOTE: The library names in MSVC are different from those in MinGW, specifically the `lib` prefix is not used for the library names.

NOTE: [GstSharp .NET bindings](#) require the MSVC binaries starting with 1.18.

NOTE: Some of the plugins shipped with the MSVC binaries link to non-gstreamer libraries built with MinGW because they are built with Autotools. [See below](#) for what this means for your application.

[Older 1.x binary releases](#) are also available.



- Home
- Features
- News
- Annual Conference
- Planet (Blogs)
- Download
- Applications
- Security Center
- GitLab
- Developers
- Documentation
- Mailing Lists
- File a Bug
- Bug Lists

- Artwork
- @gstreamer on Twitter
- @gstreamer on Mastodon

Systemeigenschaften

Computernamen Hardware **Erweitert** Computerschutz RemoteApp und Desktopverwaltung

Sie müssen als Administrator angemeldet sein, um diese Änderungen vornehmen zu können.

Leistung

Visuelle Effekte, Prozessorzeitplanung, Speichernutzung und virtuelle Hardwarespeicher

Benutzerprofile

Desktopeinstellungen bezüglich der Anmeldung

Starten und Wiederherstellen

Systemstart, Systemfehler und Debuginformationen

OK Abbrechen

Umgebungsvariablen

Umgebungsvariable bearbeiten

C:\Users\jona3814_admin\AppData\Local\Programs\Python\Python38\Scripts
C:\Users\jona3814_admin\AppData\Local\Programs\Python\Python38\python.exe
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\jona3814_admin\AppData\Local\Programs\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.29.30133\bin\amd64\x-
C:\Program Files\OpenSSL-Win64\bin
C:\gstreamer\1.0\msvc_x86_64\bin

Neu Bearbeiten Durchsuchen... Löschen Nach oben Nach unten Text bearbeiten...

OK Abbrechen

For each of the above listed targets, [a zip file will be provided](#).

If you are not sure which to pick between MSVC and MinGW, you should pick MSVC.

NOTE: The library names in MSVC are different from those in MinGW, specifically the DLLs are of the form `libname.dll.a` instead of `libname.dll`.

NOTE: [GstSharp .NET bindings](#) require the MSVC binaries starting with 1.18.

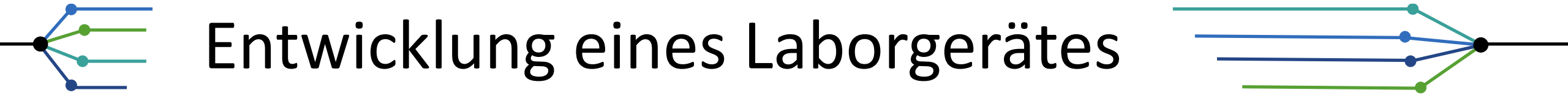
NOTE: Some of the plugins shipped with the MSVC binaries link to non-gstreamer libraries built with MinGW because they are built with Autotools. [See below](#) for what this means for your application.

[Older 1.x binary releases](#) are also available.

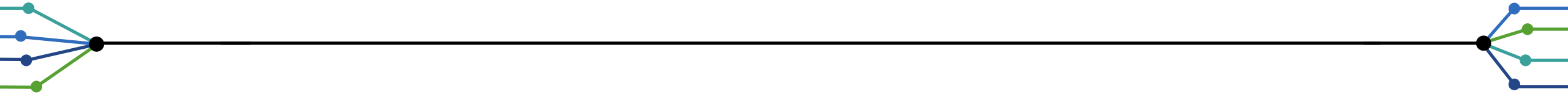
balls maintained by the GStreamer project.

ment, you will want to install both packages.

ow which may affect you based on what toolchain your app will be



Live



Ende

Alles verstanden?

CROSS Lab

