



CLOUD STORAGE & WEB COMMUNICATION APIs

CLOUD STORAGE MODELS and COMMUNICATION APIs

1. **WAMP** – AutoBahn for IoT
2. Xively Cloud for IoT
3. Python Web Application Framework – Django
4. Designing RESTful web API
5. Amazon web services for IoT
6. SkyNet IoT Messaging Platform



WAMP – AutoBahn for IoT



WAMP – AutoBahn for IoT

❑ Web Application Messaging Protocol

- ❖ Sub protocol of **Websocket**
- ❖ Provides **publish-subscribe** and **RPC messaging** patterns.
- ❖ Enables **distributed application architectures**

❖ **Key concepts of WAMP are:**

➤ Transport



- Connects **two peers**
- Default transport is **WebSocket**
- Support message based reliable **bi-directional** communication

➤ Session



- Conversation between two peers
- Runs over a Transport



WAMP – AutoBahn for IoT

Web Application Messaging Protocol

➤ Client

Peers that have one or more roles

Publish – subscriber model

RPC model

publisher

subscriber

Caller

Callee

-Publishes events to the topic maintained by the Broker

-Subscribes to the topics and receives the events including payload

Issues calls to remote procedures along with call arguments

Executes the procedures to which the calls are issued by the caller and returns the result back to caller



WAMP – AutoBahn for IoT

□ Web Application Messaging Protocol

➤ Router



- Peers that perform generic call and event routing.
- The role of the client varies as per model

publish- subscriber

RPC

Broker

Dealer

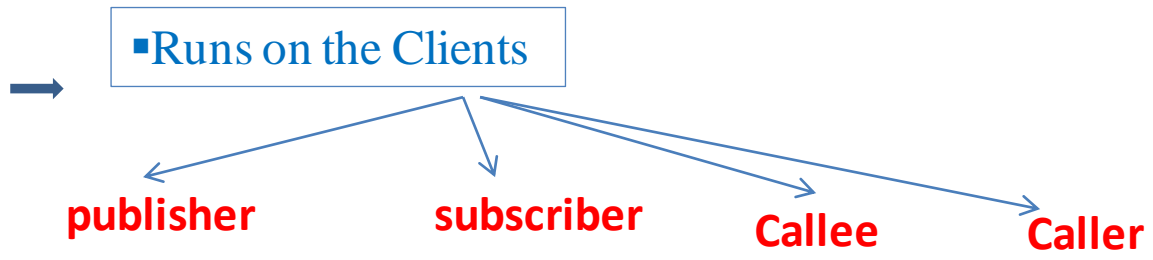
- Acts as a router
- Routes messages published to topic to all subscribers subscribed to the topic.

- Acts as a router
- Routes RPC calls from Caller to the Callee
- Routes results from Callee to the Caller.

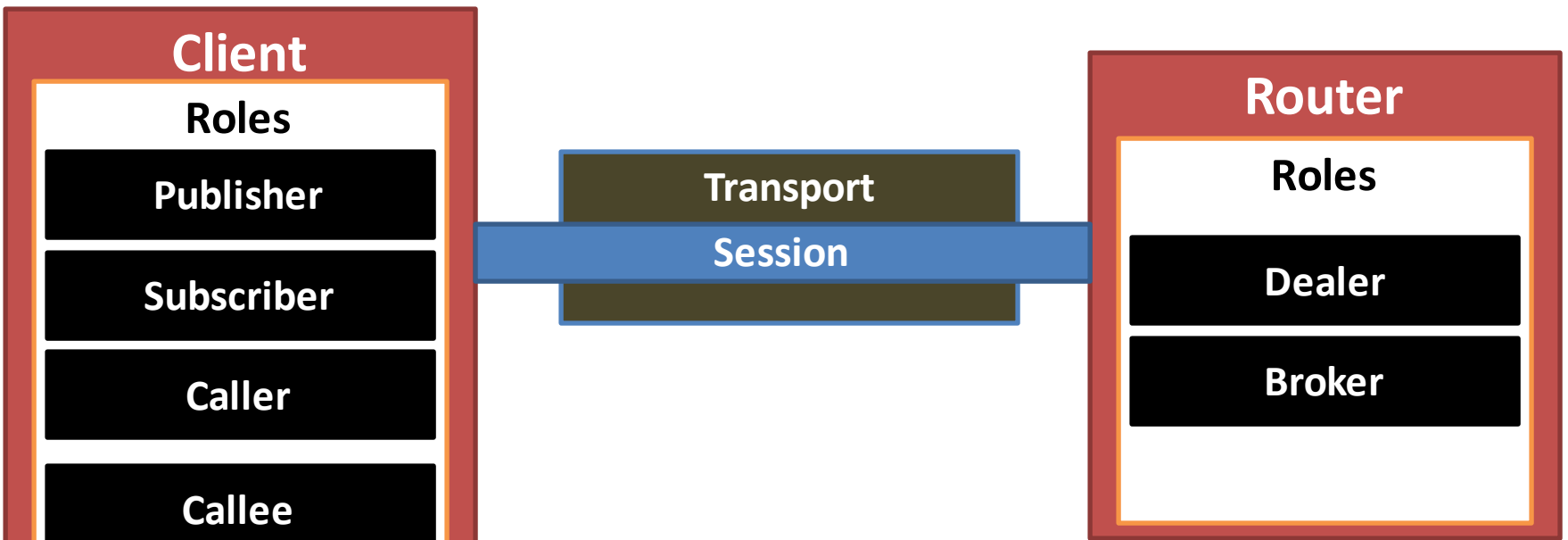


WAMP – AutoBahn for IoT

➤ Application Code

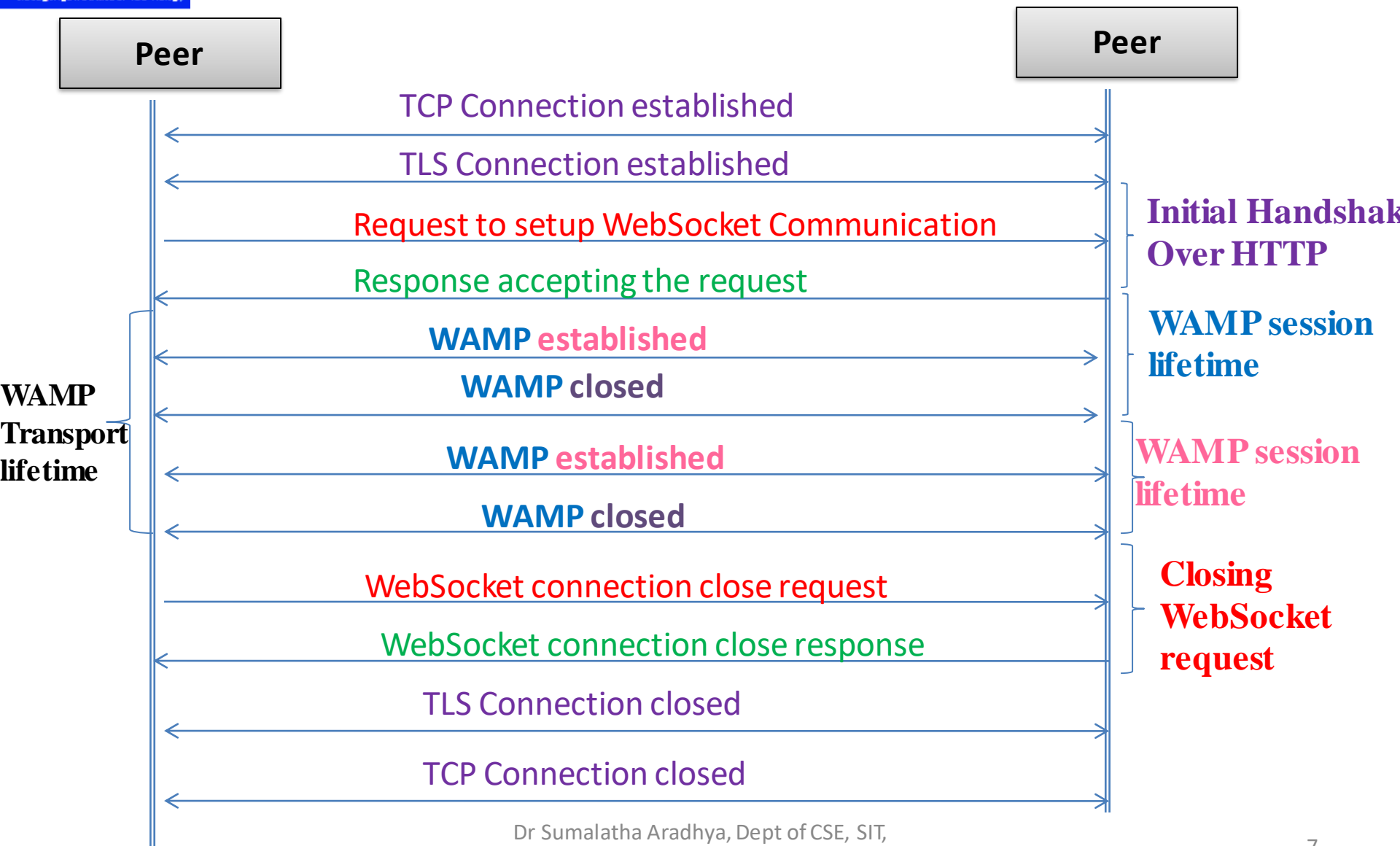


WAMP Session between Client and Router:





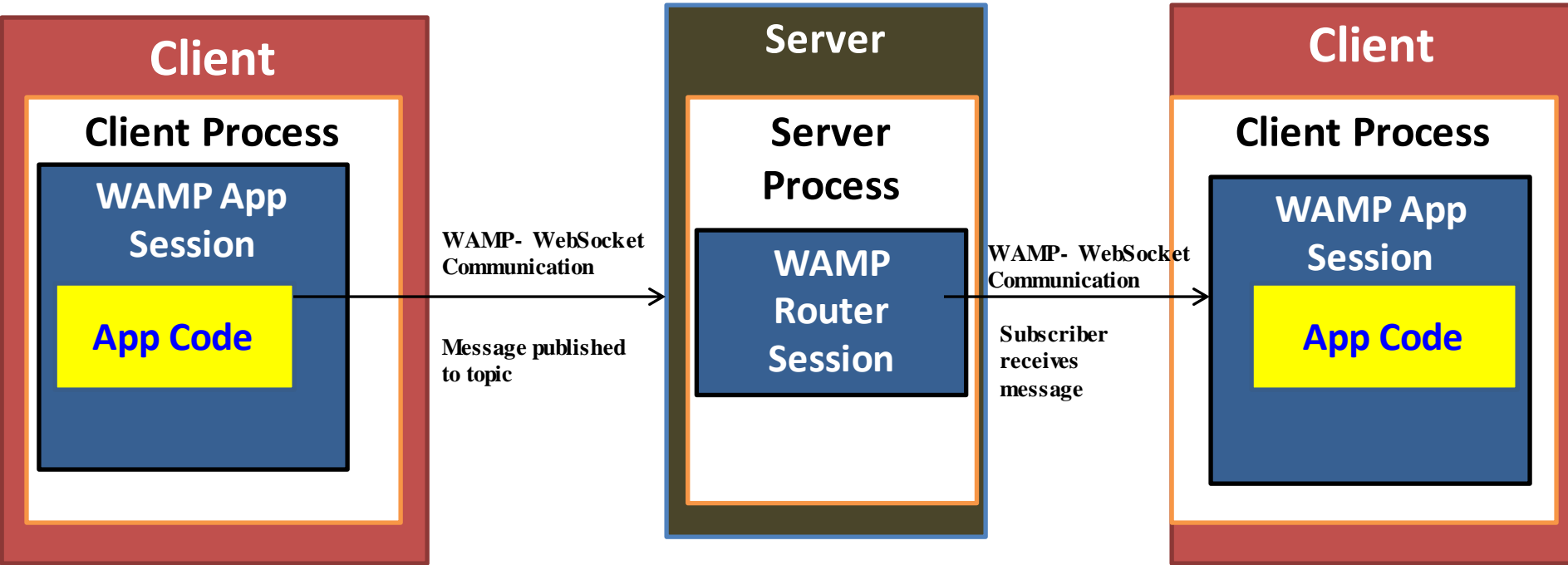
WAMP Protocol interaction between peers:





WAMP – AutoBahn for IoT

Publish – Subscribe messaging WAMP AutoBahn :



Let's look into an example of a WAMP publisher-subscriber implementation using AutoBahn:
(P.S: Follow the slides 9-)



1. Commands for Installing AutoBahn:

```
sudo apt-get install python-twisted python-dev
sudo apt-get install python-pip
sudo pip install --upgrade twisted
sudo pip install --upgrade autobahn
```

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>pip install autobahn
Collecting autobahn
  Downloading https://files.pythonhosted.org/packages/a6/56/c218e6ffadbc42a5e0ba8a837a6cc57fa2dd98ab8350e5097a724e7abfd7/autobahn-20.4.3-py2.py3-none-any.whl (1.1MB)
  |-----| 1.1MB 819kB/s
Collecting txai0>=20.3.1 (from autobahn)
  Downloading https://files.pythonhosted.org/packages/4f/82/0cd8d81d57e55a598cd4cef10c6e971dbcaf437e4f138dc1624cf7c1388e/txai0-20.4.1-py2.py3-none-any.whl
Collecting cryptography>=2.7 (from autobahn)
  Downloading https://files.pythonhosted.org/packages/f7/00/33abf1278178dd4314fd943a8b25a39d370ach151e72b0299975af478d1c/cryptography-2.9.2-cp38-cp38-win_amd64.whl (1.5MB)
  |-----| 1.5MB 6.4MB/s
Collecting six>=1.4.1 (from cryptography>=2.7->autobahn)
  Downloading https://files.pythonhosted.org/packages/65/eb/1f97cb97bfc2390a276969c6fae16075da282f5058082d4cb10c6c5c1dba/six-1.14.0-py2.py3-none-any.whl
Collecting cffi!=1.11.3, >=1.8 (from cryptography>=2.7->autobahn)
  Downloading https://files.pythonhosted.org/packages/b4/78/eefed9f51baa2b2c891915b4590a237ed6bc6d02beec177fc503bae8eb53/cffi-1.14.0-cp38-cp38-win_amd64.whl (177kB)
  |-----| 184kB 6.4MB/s
Collecting pycparser (from cffi!=1.11.3, >=1.8->cryptography>=2.7->autobahn)
  Downloading https://files.pythonhosted.org/packages/ae/e7/d9c3a176ca4b02024deb f82342dab36efadfc5776f9c8db077e8f6e71821/pycparser-2.20-py2.py3-none-any.whl (112kB)
  |-----| 112kB 2.2MB/s
Installing collected packages: txai0, six, pycparser, cffi, cryptography, autobahn

WARNING: The scripts wamp.exe and xbrnetwork.exe are installed in 'c:\users\cclab\appdata\local\programs\python\python38\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location

Successfully installed autobahn-20.4.3 cffi-1.14.0 cryptography-2.9.2 pycparser-2.20 six-1.14.0 txai0-20.4.1
```



2. Clone AutoBahn Python from github:

```
git clone https://github.com/tavedo/AutobahnPython.git
```

P.S: this link
Doesn't exist now

Alternative link

```
C:\Windows\system32>git clone https://github.com/tavedo/AutobahnPython.git
Cloning into 'AutobahnPython'...
remote: Repository not found.
fatal: repository 'https://github.com/tavedo/AutobahnPython.git/' not found
```

```
C:\Windows\system32>git clone https://github.com/crossbario/autobahn-python.git
Cloning into 'autobahn-python'...
remote: Enumerating objects: 75, done.
remote: Counting objects: 100% (75/75), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 27936 (delta 40), reused 34 (delta 17), pack-reused 27861
Receiving objects: 100% (27936/27936), 16.24 MiB | 4.60 MiB/s, done.
Resolving deltas: 100% (18974/18974), done.
C:\Windows\system32>
```

```
(base) C:\Users\CEDLAB\anaconda3\envs>git clone https://github.com/pawelnhm/Auto
bahnPython.git
Cloning into 'AutobahnPython'...
remote: Enumerating objects: 10007, done.
remote: Total 10007 (delta 0), reused 0 (delta 0), pack-reused 10007
Receiving objects: 100% (10007/10007), 3.56 MiB | 1.09 MiB/s, done.
Resolving deltas: 100% (6180/6180), done.
```



❖ 3. Create WAMP publisher component

Publisher component: Publishes a message containing the current time stamp to a topic named **'test-topoc'**

```
D:\Suma\Online Classes Reference\IoT\PublisherApp.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Window ?
Categories.php x Admins.php x AddNewPost.php x Login.php x FullPost.php x Comments.php x autoBahn.sh x PublisherApp.py x
1 from twisted.internet import reactor
2 from twisted.internet.defer import inlinecallbacks
3 from autobahn.twisted.util import sleep
4 from autobahn.twisted.wamp import ApplicationSession
5 import time,datetime
6
7 def getData():
8     #Generate message
9     timestamp = datetime.datetime.fromtimestamp(time.time()).strftime('%Y-%m-%d%H:%M:%S')
10    data = "Message at time-stamp: " +str(timestamp)
11    return data
12
13 #An application component that publishes an event every second
14 class Component(ApplicationSession):
15     @inlinecallbacks
16     def onJoin(self,details):
17         while True:
18             data = getData()
19             self.publish('test-topoc',data)
20             yield sleep(1)
21
```



4. Create WAMP Subscriber component

Subscriber Component: Subscribers to the **'test-topic'**.

```
D:\Suma\Online Classes Reference\IoT\subscriberApp.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Window ?
Categories.php Admins.php AddNewPost.php Login.php FullPost.php Comments.php autoBahn.sh PublisherApp.py subscriberApp.py
1 from twisted.internet import reactor
2 from twisted.internet.defer import inlinecallbacks
3 from autobahn.twisted.wamp import ApplicationSession
4
5
6 #An application component that publishes an event every second
7 class Component(ApplicationSession):
8     @inlinecallbacks
9     def onJoin(self, details):
10         self.received = 0
11
12         def on_event(data):
13             print("Received message: " +data)
14             yield self.subscribe(on_event, 'test-topic')
15
16     def onDisconnect(self):
17         reactor.stop()
18
```



5. Run the application router on a WebSocket transport server

i) Run the application router on a WebSocket transport server:

```
python AutobahnPython/examples/twisted/wamp/basic/server.py
```

Example,



```
(base) C:\Users\CEDLAB\anaconda3\envs\AutobahnPython\examples\twisted\wamp\auth\persona>python server.py
```

ii) Run the Publisher component on a WebSocket transport client:

```
python AutobahnPython/examples/twisted/wamp/basic/client.py --component "PublisherApp.Component"
```



```
(base) C:\Users\CEDLAB\anaconda3\envs\AutobahnPython\examples\twisted\wamp\basic>python client.py --component "PublisherApp.Component"
```

iii) Run the Subscriber component on a WebSocket transport client:

```
python AutobahnPython/examples/twisted/wamp/basic/client.py --component "subscriberApp.Component"
```

```
(base) C:\Users\CEDLAB\anaconda3\envs\AutobahnPython\examples\twisted\wamp\basic>python client.py --component "subscriberApp.Component"
```



Exercise : Explore the WAMP autobahn

References: <https://crossbario.com/blog/Small-Pragmatic-WAMP-Demo/>
<https://github.com/pawelmhm/AutobahnPython>
<https://github.com/claws/AutobahnPython>