

The background of the slide features a scenic view of a wooden pier made of weathered planks, extending from the bottom center into a vast, calm blue ocean. The sky above is filled with large, billowing clouds that are bathed in the warm, golden light of either a sunrise or sunset, creating a vibrant palette of orange, yellow, and pink against the deep blue of the upper sky.

**SCS Standardization and Certification**

**Felix Kronlage-Dammers**

**fkr@osb-alliance.com**

One Platform -  
***standardised***, built and  
operated by many.



**SPRIN-D**



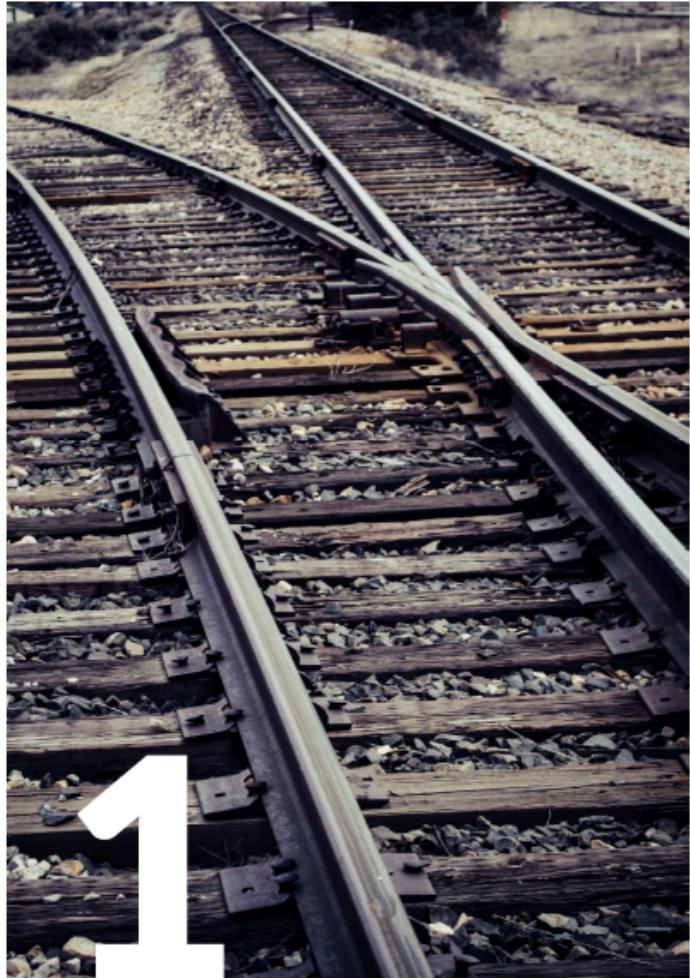
Gefördert durch:



Bundesministerium  
für Wirtschaft  
und Klimaschutz

aufgrund eines Beschlusses  
des Deutschen Bundestages

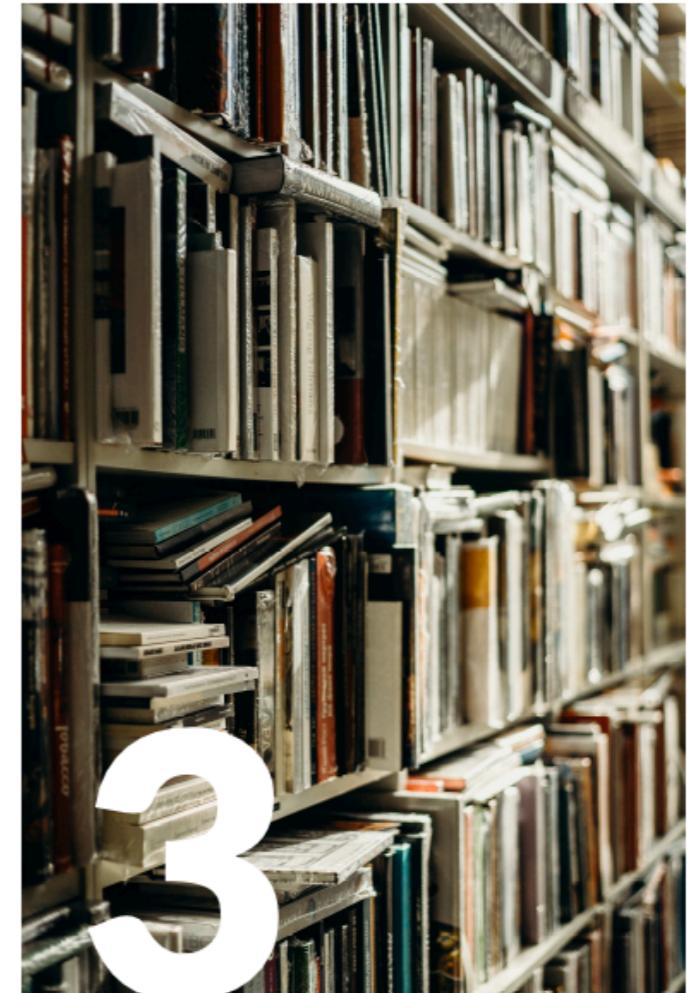
# Sovereign Cloud Stack Deliverables



Certifiable Standards



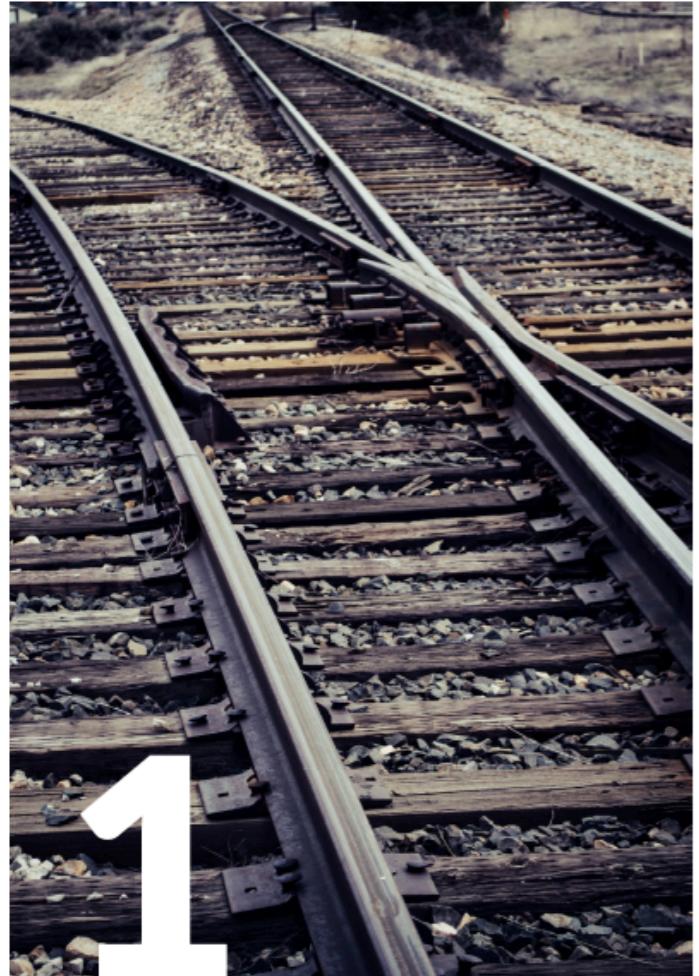
Modular Open Source  
Reference Implementation



Operational Knowledge

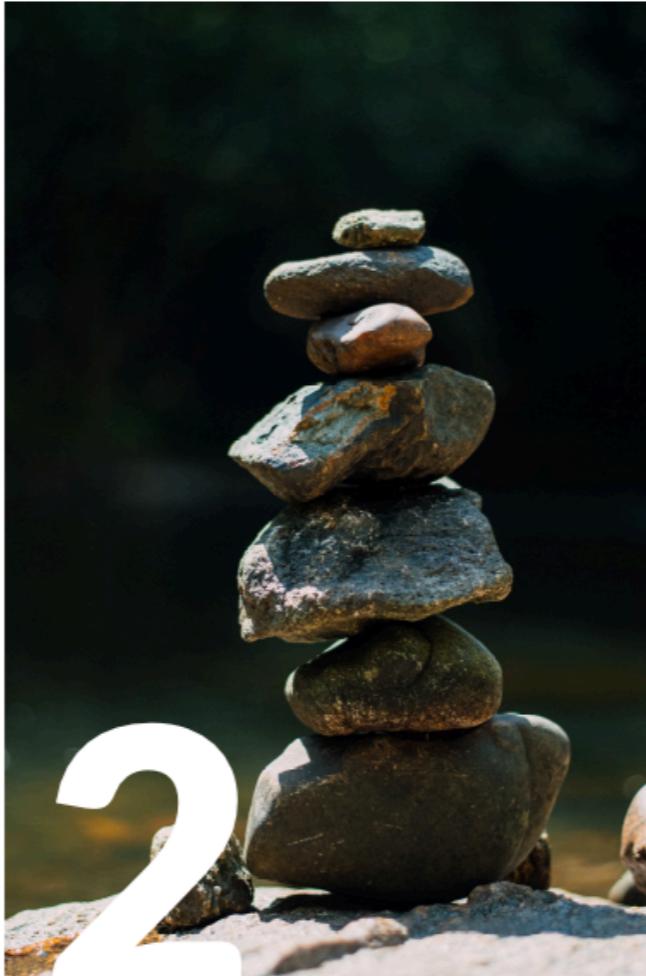


# Sovereign Cloud Stack Deliverables



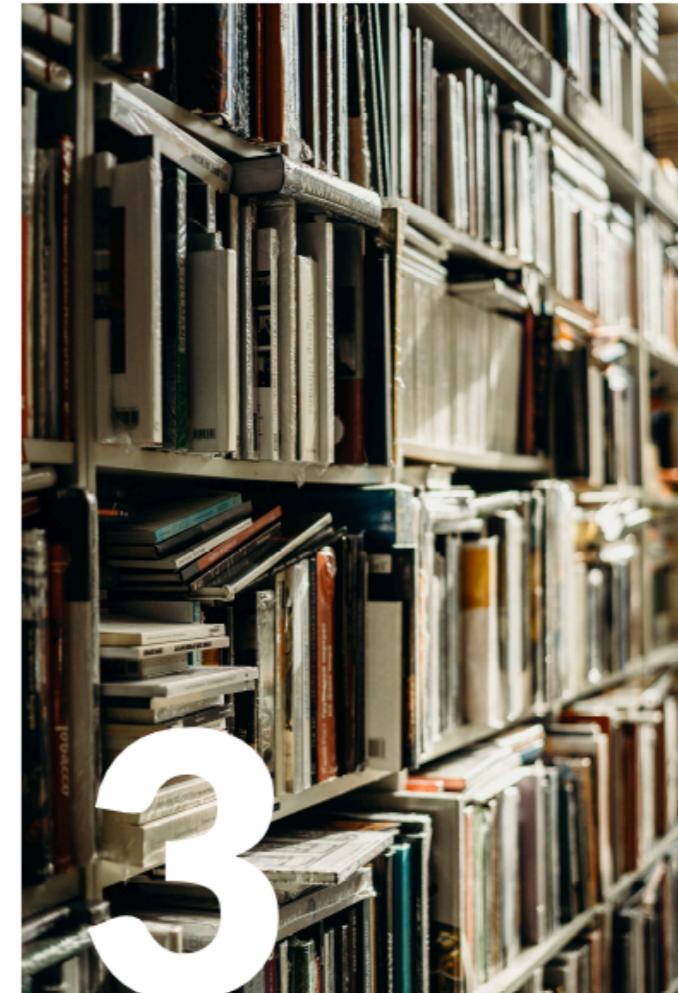
1

Certifiable Standards



2

Modular Open Source  
Reference Implementation



3

Operational Knowledge

**Why strive for standards?**

# Interoperability is awesome



# Interoperability is awesome



```
157     def get_server_external_ipv4(cloud, server):
158         """Find an externally routable IP for the server.
159
160         ...
161
162         * Cloud has externally routable IP from neutron but neutron APIs don't
163           work (only info available is in nova server record) (rackspace)
164         * Cloud has externally routable IP from neutron (runabove, ovh)
165         * Cloud has externally routable IP from neutron AND supports optional
166           private tenant networks (vexxhost, unitedstack)
167         * Cloud only has private tenant network provided by neutron and requires
168           floating-ip for external routing (dreamhost, hp)
169         * Cloud only has private tenant network provided by nova-network and
170           requires floating-ip for external routing (auro)
171
172         :param cloud: the cloud we're working with
173         :param server: the server dict from which we want to get an IPv4 address
174         :return: a string containing the IPv4 address or None
175         """
176
177         # If the server has an external IP, return it
178         if 'external_ip' in server:
179             return server['external_ip']
180
181         # If the server has an IP assigned via Nova's private network, return it
182         if 'fixed_ips' in server:
183             for fixed_ip in server['fixed_ips']:
184                 if fixed_ip['version'] == 4:
185                     return fixed_ip['ip_address']
186
187         # If the server has an IP assigned via Neutron's private network, return it
188         if 'addresses' in server:
189             for network_name in server['addresses']:
190                 if network_name == 'private':
191                     for ip in server['addresses'][network_name]:
192                         if ip['version'] == 4:
193                             return ip['addr']
194
195         # If the server has an IP assigned via Neutron's public network, return it
196         if 'addresses' in server:
197             for network_name in server['addresses']:
198                 if network_name == 'public':
199                     for ip in server['addresses'][network_name]:
200                         if ip['version'] == 4:
201                             return ip['addr']
202
203         # If none of the above, return None
204         return None
```

**Standards will turn "the box of Legos" into a coherent product**

# DIGITAL SOVEREIGNTY



## DIGITAL SOVEREIGNTY EVERYWHERE

# ... more than mere legal compliance



Dimension of competence

Technological dimension

Freedom of choice

Legal dimension



## SCS: Why standardization?

- Real choice (2nd dimension DigiSov) requires lock-in-less choice
  - Technically fully technically compatible providers available
  - Self-Hosting fully compatible infrastructure must be realistic
- „Virtual Hyperscaler“ vision
  - Users can leverage many clouds as one
  - Requires common feature set, common APIs, common system behavior (baseline)
  - Requires user federation
- Enables joint development, joint operational practices

# ... more than mere legal compliance



Dimension of competence

Technological dimension

Freedom of choice

Legal dimension

# Sovereignty & SCS certification

## Levels of digital sovereignty

4: Operational transparency and knowledge available (skills building)

3: Technological transparency and capability to contribute & shape

2: Choice b/w many providers, in-sourcing (on-prem) option

1: Legal compliance (GDPR)

0: None



## SCS certification levels



4: “**SCS**-sovereign” – Ops/IAM stacks OSS as well, transparency on monitoring, incidents, contributing to OpenOperations (5 x open)

3: “**SCS**-open” – SBOM for functional stack available and fully OSS (4 x open)

2: “**SCS**-compatible” – technical compatibility (conformance tests pass: CNCF, OIF, SCS)

1: None (rely on ENISA / Gaia-X labels / Law)

**Who benefits from standards?**

# Compliant cloud environments

This is a list of clouds that we test on a nightly basis against the certificate scope SCS-compatible IaaS.

Name	Description	Operator	SCS-compatible IaaS	HealthMon
gx-scs	Dev environment provided for SCS & GAIA-X context	plusserver GmbH	v4, v5*	HM
aov.cloud	Community cloud for customers	aov IT.Services GmbH	-	HM
CNDS	Public cloud for customers	artcodix GmbH	v4, v5*	HM
pluscloud open (4 regions)	Public cloud for customers	plusserver GmbH	prod1:  v4, v5* prod2:  v4, v5* prod3:  v3++ prod4:  v4, v5*	HM1 HM2 HM3 HM4
PoC KDO	Cloud PoC for FITKO	KDO Service GmbH / OSISM GmbH	v4, v5*	(soon)
PoC WG-Cloud OSBA	Cloud PoC for FITKO	Cloud&Heat Technologies GmbH	v4, v5*	HM
REGIO.cloud	Public cloud for customers	OSISM GmbH	v4, v5*	HM
sysEleven (2 SCS regions)	Public OpenStack Cloud	SysEleven GmbH	dus2:  v3++ ham1:  v3++	(soon) (soon)
Wavestack	Public cloud for customers	noris network AG/Wavecon GmbH	-	HM

# SCS-compatible IaaS

Scope versions ->	v3	v4	v5
State	Deprecated	Effective	Draft
Stabilized at	2024-02-28	2024-02-28	
<b>Modules</b>			
<a href="#">OpenStack Powered Compute v2022.11</a>	X	X	X
<a href="#">scs-0100-v3.1: Flavor naming v3.1</a>	X	X	X
<a href="#">scs-0101-v1: Entropy v1</a>		X	X
<a href="#">scs-0102-v1: Image metadata v1</a>	X	X	X
<a href="#">scs-0103-v1: Standard flavors</a>		X	X
<a href="#">scs-0104-v1: Standard images</a>		X ( <a href="#">image_spec</a> )	X ( <a href="#">image_spec</a> )

Standard	Description
<a href="#">scs-0100</a>	SCS Flavor Naming Standard
	Supplement: Implementation and Testing Notes
<a href="#">scs-0101</a>	SCS Entropy
	Supplement: Implementation and Testing Notes
<a href="#">scs-0102</a>	SCS Image Metadata Standard
	Supplement: SCS Image Metadata: Implementation and Testing Notes
<a href="#">scs-0103</a>	SCS Standard Flavors and Properties
<a href="#">scs-0104</a>	SCS Standard Images
	Supplement: Implementation Notes
<a href="#">scs-0110</a>	SSD Flavors
<a href="#">scs-0111</a>	Decisions for the Volume Type Standard
<a href="#">scs-0112</a>	SONiC Support in SCS
<a href="#">scs-0113</a>	Security Groups Decision Record
<a href="#">scs-0114</a>	Volume Type Standard
<a href="#">scs-0115</a>	Default Rules for Security Groups
<a href="#">scs-0116</a>	Key Manager Standard
	Supplement: SCS Key Manager Standard: Implementation and Testing Notes
<a href="#">scs-0117</a>	Volume Backup Functionality
<a href="#">scs-0118</a>	Taxonomy of Failsafe Levels
	Supplement: SCS Taxonomy of Failsafe Levels: Examples of Failure Cases and their impact on IaaS and KaaS resources
<a href="#">scs-0119</a>	Replacement of the deprecated ceph-ansible tool
<a href="#">scs-0120</a>	Cluster-API images
<a href="#">scs-0121</a>	Availability Zones Standard
	Supplement: SCS Availability Zone Standard: Implementation and Testing Notes

Standard	Description	Draft	Effective	Deprecated*
scs-0100	SCS Flavor Naming Standard	-	v3	v1, v2
	Supplement: Implementation and Testing Notes	w1	-	-
scs-0101	SCS Entropy	-	v1	-
	Supplement: Implementation and Testing Notes	w1	-	-
scs-0102	SCS Image Metadata Standard	-	v1	-
	Supplement: SCS Image Metadata: Implementation and Testing Notes	w1	-	-
scs-0103	SCS Standard Flavors and Properties	-	v1	-
scs-0104	SCS Standard Images	-	v1	-
	Supplement: Implementation Notes	w1	-	-
scs-0110	SSD Flavors	-	v1	-
scs-0111	Decisions for the Volume Type Standard	v1	-	-

scs-0112	SONiC Support in SCS	v1	-	-
scs-0113	Security Groups Decision Record	v1	-	-
scs-0114	Volume Type Standard	v1	-	-
scs-0115	Default Rules for Security Groups	v1	-	-
scs-0116	Key Manager Standard	v1	-	-
	Supplement: SCS Key Manager Standard: Implementation and Testing Notes	w1	-	-
scs-0117	Volume Backup Functionality	v1	-	-
scs-0118	Taxonomy of Failsafe Levels	v1	-	-
	Supplement: SCS Taxonomy of Failsafe Levels: Examples of Failure Cases and their impact on IaaS and KaaS resources	w1	-	-
scs-0119	Replacement of the deprecated ceph-ansible tool	v1	-	-
scs-0120	Cluster-API images	v1	-	-
scs-0121	Availability Zones Standard	v1	-	-

# SCS-compatible KaaS

Scope versions ->	v1	v2
State	Draft	Draft
Stabilized at		
Modules		
CNCF Kubernetes conformance		X
scs-0210-v2: Kubernetes version policy	X	X
scs-0214-v2: Kubernetes node distribution and availability	X	X

# KaaS Standards

Standard	Description	Draft	Effective	Deprecated*
scs-0200	Using Sonobuoy for KaaS conformance tests	v1	-	-
scs-0210	SCS K8S Version Policy	-	v2	v1
	Supplement: Implementation and Testing Notes	w1	-	-
scs-0211	SCS KaaS default storage class	v2	v1	-
	Supplement: Implementation and Testing Notes	w1	-	-
scs-0212	Requirements for container registries	v1	-	-
scs-0213	Kubernetes Nodes Anti Affinity	v1	-	-
scs-0214	Kubernetes Node Distribution and Availability	v2	v1	-
	Supplement: Implementation and Testing Notes	w1	-	-
scs-0215	Robustness features for Kubernetes clusters	v1	-	-
scs-0216	Requirements for testing cluster-stacks	v1	-	-
scs-0217	Kubernetes cluster hardening	v1	-	-
scs-0218	Container registry for SCS standard implementation	v1	-	-

# From talk at EOF event - Pt. 1

## Notable standards I

- *notably un-exciting:*
  - improving discoverability
    - harmonizing flavor extra\_specs, image metadata, image sources
  - or working around bad discoverability by restricting your freedom 😢
    - harmonizing flavor names, volume type descriptions

# From talk at EOF event - Pt. 2

## Notable standards II

- exciting: standardizing functionality and robustness guarantees 😍
  - mandatory and supported APIs
  - encryption
  - update cycles
  - ... this is mostly work in progress though

# scs-0101: SCS Entropy

The SCS-0101 Entropy Standard ensures adequate entropy is available in virtual instances, crucial for operations such as secure key creation in cryptography. The standard recommends using kernel version 5.18 or higher and activating the `hw_rng_model: virtio` attribute for images, while compute nodes should employ CPUs with entropy accessing instructions unfiltered by the hypervisor. It allows the infusion of the hosts entropy sources into virtual instances and ensures the availability and quality of entropy in virtual environments, promoting system security and efficiency.

Version	Type	State	stabilized	deprecated
<a href="#">scs-0101-v1</a>	Standard	Stable	2024-02-08	-

# Requirements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

In addition, "FORBIDDEN" is to be interpreted equivalent to "MUST NOT".

# Sovereign Cloud Standard documents

One of the main products of the SCS organisation are Sovereign Cloud Standard documents.

## Types of documents

### Procedural

A procedural SCS document describes a process, a policy or a guideline to which the SCS community adheres.

### Standard

A standard SCS document describes a technical standard for SCS compliant clouds. Note that it may not be necessary for all clouds to implement all standards.

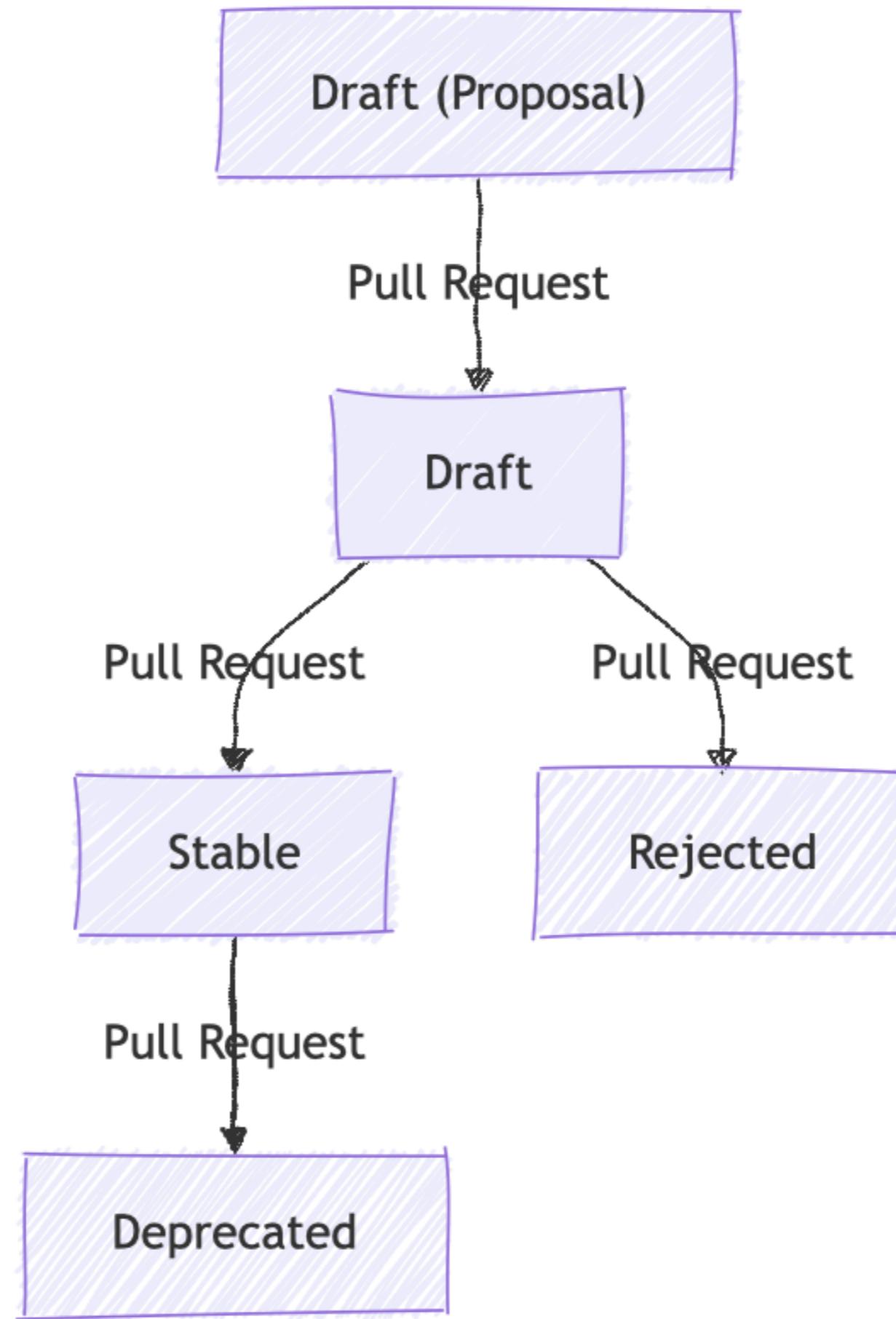
### Decision Record

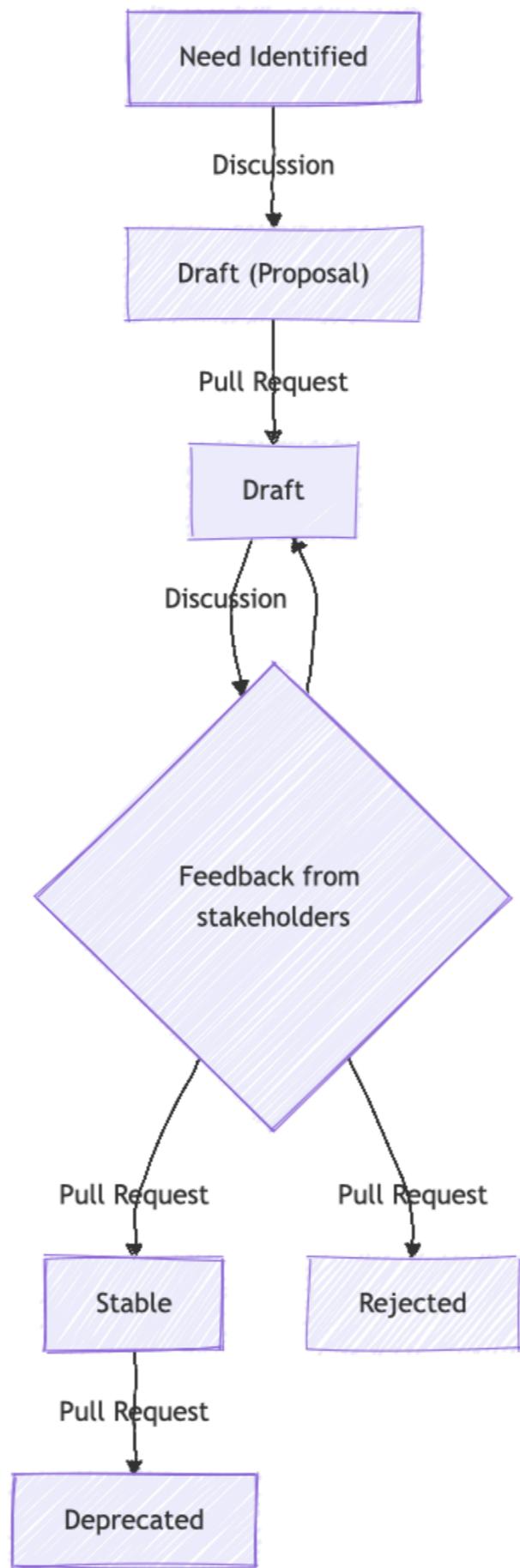
Sometimes during the development of the SCS standard, a complex technical decision needs to be taken, which does not directly result in a new standard.

The SCS document format formally integrates the documentation of such decisions as documents of type [Decision Record](#).

### Supplement

A supplement extends a Standard with additional information, such as implementation and testing notes, that is merely informative, but not authoritative, and that may be subject to change more frequently than the standard itself.





„We believe in  
collaboration  
rather than  
competition!“

September 2022

ec  
os

**Building a network of communities...**

# ...together with ALASCA

## Collaboration on Cloud Standards



Kurt Garloff, Marius Feldmann  
November 27, 2023

## Collaboration on Cloud Standards

**Sovereign Cloud Stack and ALASCA e.V.: joining forces for good standards**

# ...Upstream at OpenInfra

## PublicCloudSIG

---

**Status:** Active

**Chairs:**

- \* Tobias Rydberg <tobias.rydberg@cleura.com>
- \* Felix Kronlage-Dammers <fkr@osb-alliance.com>

The aim of this group is to represent the interests of the OpenStack public cloud provider community, and to further adoption of OpenStack public cloud usage.

# ...with the public hand

IG BvC

Die IG BvC stellt sich vor ...

## IG BvC (Interessengemeinschaft Betrieb von Containern)



Die IG Betrieb von Containern (IG BvC) ist ein Zusammenschluss von Datenzentralen, Softwarelieferanten und Organisationen der Öffentlichen Verwaltung.

# Adoption



# Costs of making an OpenStack cluster SCS-compliant



Hannes Baum, Martin Morgenstern

May 13, 2024

Have you ever wondered how much effort it would take to adopt SCS standards in your OpenStack cloud? We wanted to know this too, and as part of our work in the SCS standards team, we evaluated the process of making a vanilla OpenStack cluster SCS-compliant. In this blog post, we want to share the results of our findings and the process we went through. Rest assured – it is actually quite easy to adopt SCS standards!

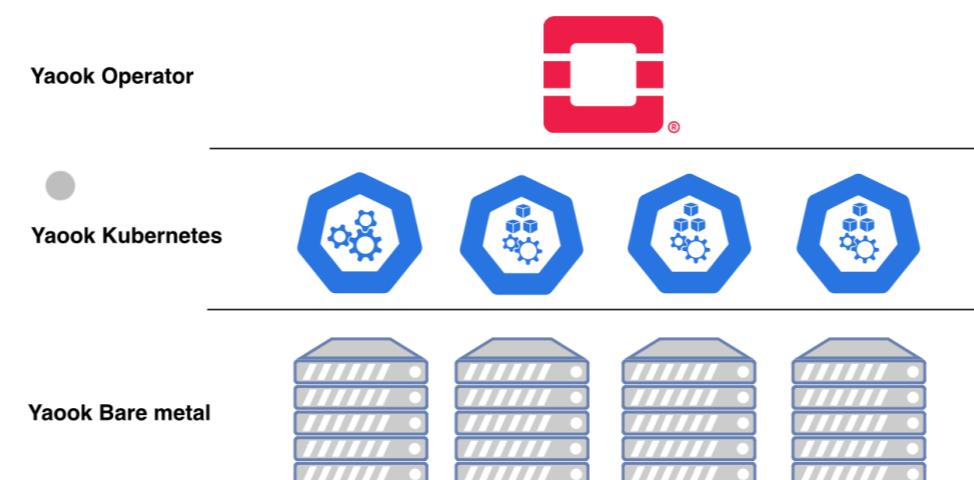
## Where we started from

Our focus in this evaluation was on OpenStack clusters and therefore the IaaS standards, because for the IaaS layer we already had a reference *SCS Compatible IaaS* scope at the time we started (in the future, a similar evaluation and blog post for the KaaS layer is planned).

For the purpose of our evaluation, we set up two OpenStack clusters with [Yaook](#) ("Yet Another OpenStack On Kubernetes"): a virtualized test setup in our OpenStack cloud – i.e., OpenStack in OpenStack – and a bare-metal production setup.

Yaook is a lifecycle management tool for OpenStack which leverages a Kubernetes cluster (provided by [Yaook K8s](#)) to deploy and manage OpenStack components by means of the [Yaook Operator](#). For the bare-metal production deployment we additionally used [Yaook Bare Metal](#) to deploy and manage server hardware, including rollout and configuration of operating systems, networks and disks.

This test setup is represented in the following visualization provided in the Yaook documentation:



At the time of writing, a vanilla Yaook deployment is not SCS-compliant and, as such, it is the ideal playground for our evaluation. Even better: the lessons we learned while adopting IaaS standards in these deployments can be easily transferred to other OpenStack deployments which do not use Yaook.

## Required standards

As it was already explained above, the main effort leading to this post was focused on the IaaS standards, mainly because it was clearer which standards needed to be fulfilled for *SCS Compatible IaaS* scope. In the SCS standardization framework, a *scope* groups multiple SCS standards for a certain layer (e.g., IaaS) to provide a common ground for certification of a cloud

# **SCS is standardized?**

- 2022 was the year of the reference implementation
- 2023 was supposed to become the year with strong focus on standards
- 2024 became the year with the strong focus on standards

## SCS Standardization and Certification - a Deep Dive

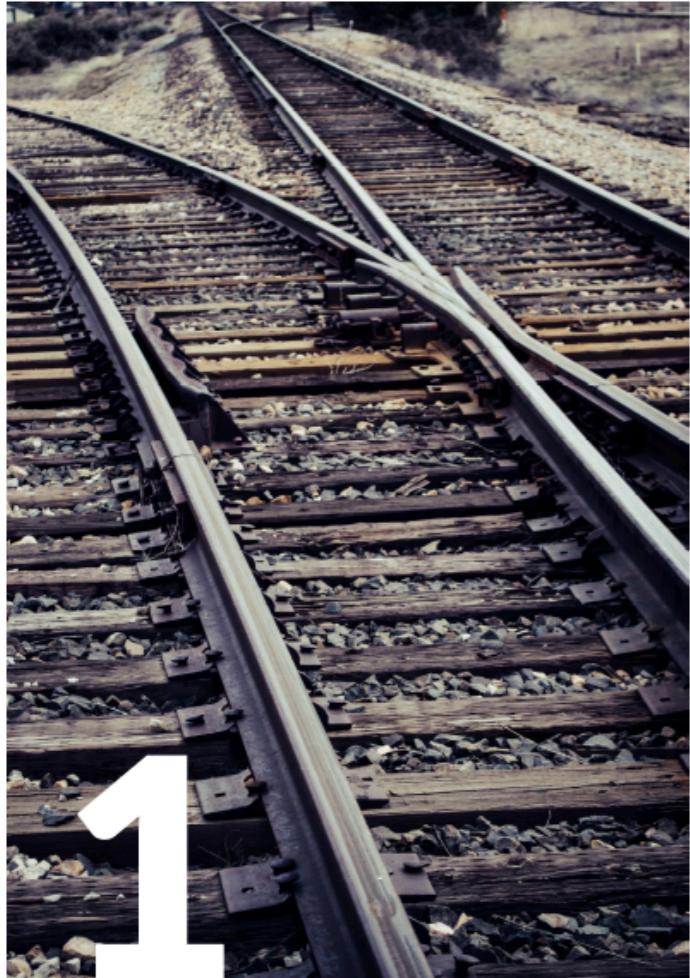
-

The Sovereign Cloud Stack (SCS) project was started three and half years ago with the vision to standardize cloud infrastructure and offerings and change how clouds are operated and offered for the better.

Sometime into the project the statement „One platform – standardized, built and operated by many.“ was found to describe what we (the project) envision.

The (initial) funding of the project will end this September and we want to give an honest and sincere overview of what we have achieved so far, the impact we've made on our surrounding ecosystems and will elaborate on why this project needs to continue and how the leverage will be in the future with a focus on the standardization and certification. Specifically, the talk will dive deep into the way the standards are developed and adapted in the industry – ranging from CSPs that build their cloud on top of the SCS reference implementation, CSPs that adapted the standards in their existing environments up to other software solutions such as Yaook who adapted the SCS standards in their deployment mechanisms.

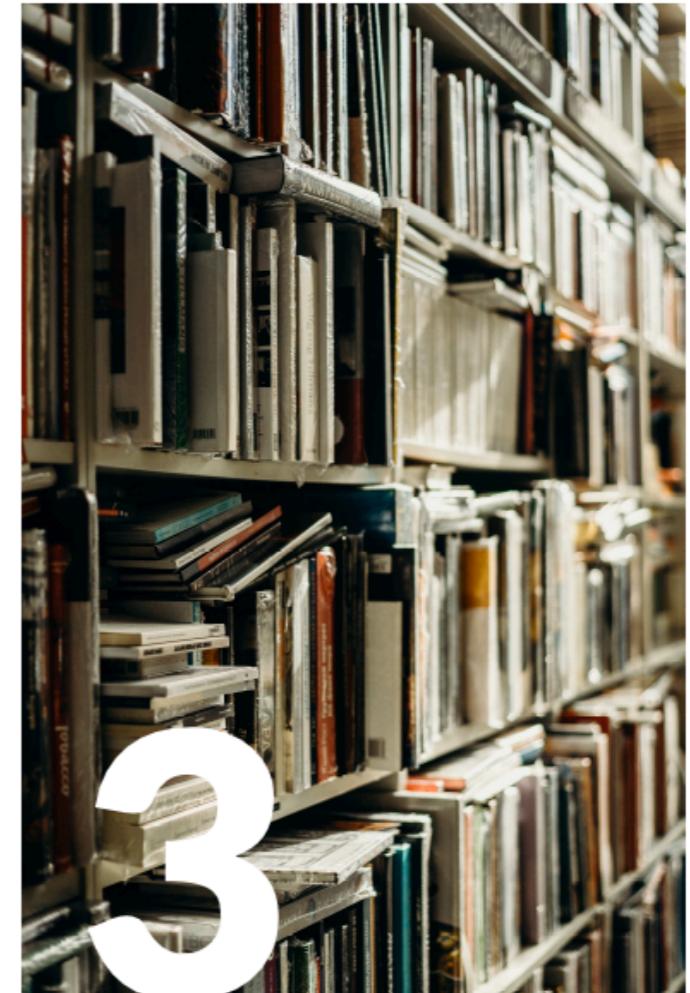
# Sovereign Cloud Stack Deliverables



Certifiable Standards



Modular Open Source  
Reference Implementation

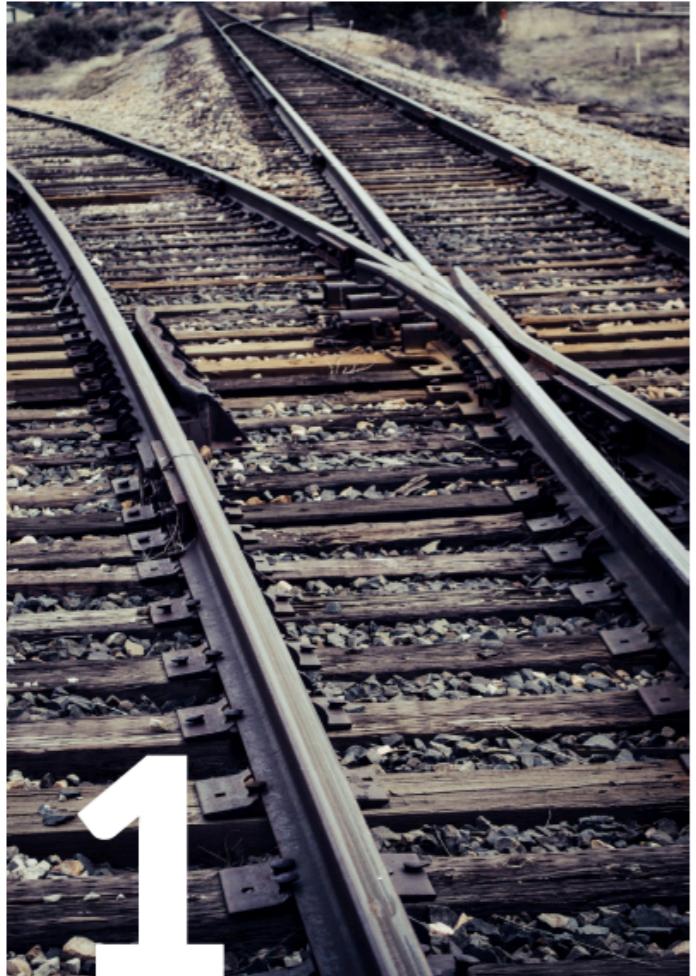


Operational Knowledge



# Sovereign Cloud Stack Deliverables

## Forum SCS-Standards



Certifiable Standards



Modular Open Source  
Reference Implementation



Operational Knowledge



---

*The Forum SCS-Standards ensures that the continuous development of the SCS Standards and the certification rules and measures is guaranteed and takes place in an open, transparent process.*

*The Forum SCS-Standards sees itself as a committee in which the rules of the SCS community are jointly discussed and decided.*

---

# Founding members of *Forum SCS-Standards*

A scenic landscape featuring a paved road that curves gently to the right, leading into the distance. The road is flanked by lush green grass and wildflowers. On either side of the road are fields, with numerous wind turbines standing tall in the background under a clear blue sky. The overall atmosphere is one of a rural, sustainable environment.

- artcodix
- Cloud&Heat
- dNation
- plusserver
- secunet
- SysEleven
- Wavecon
- b1-systems
- Dataport
- OSISM
- ScaleUp
- stackXperts
- Sysself
- Yorizon

# Sovereign Cloud Stack Deliverables

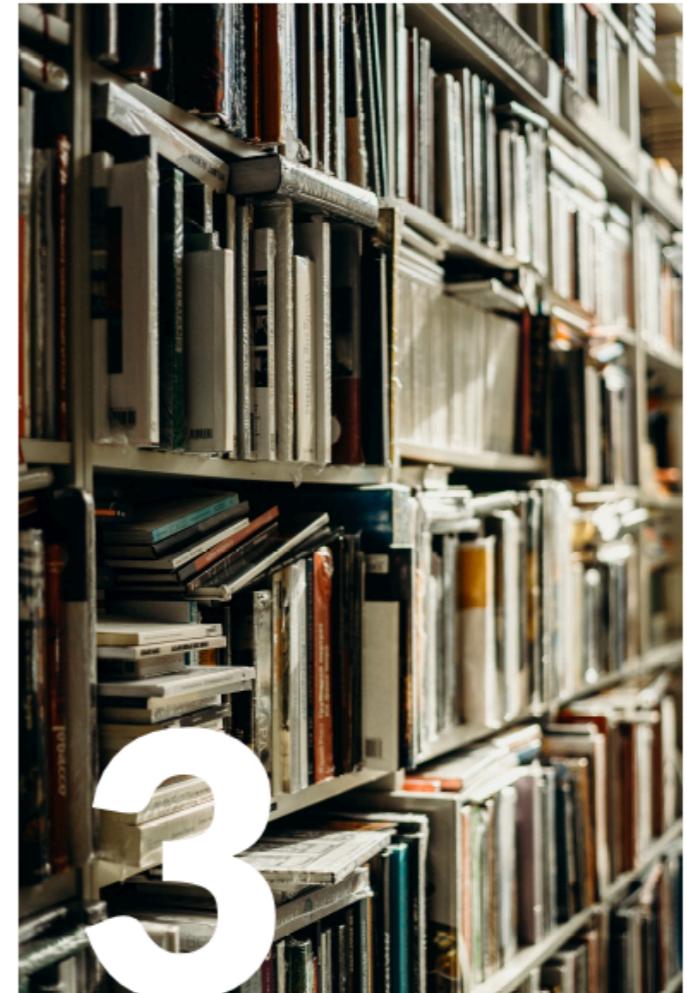


Certifiable Standards



2

Modular Open Source Reference Implementation



Operational Knowledge

# Foster the community

- Project Board
- Software-Stacks need to be commercially viable
  - also in order to sustain a healthy upstream

# the tip of the iceberg

artificial-intelligence  
piobig2871 alexander-diab  
gtema tonifinger garloff  
tsmado markus-hentschkitsudaiki  
mbuechse cah-hbaum chess-knight  
bitkeks kgube anjastrunk martinmo  
michal-gubricky josephineSei fraugabel  
cah-patrickthiem fdobrovolny  
shmelkin FriederikeZelke  
jschoone

**Thank you ALASCA** 😊

# **Follow Up ...with me**

- fkr@osb-alliance.com
- <https://matrix.to/#/@fkronlage:matrix.org>

**...or the project**

- <https://scs.community/>
- <https://matrix.to/#/#scs-general:matrix.org>