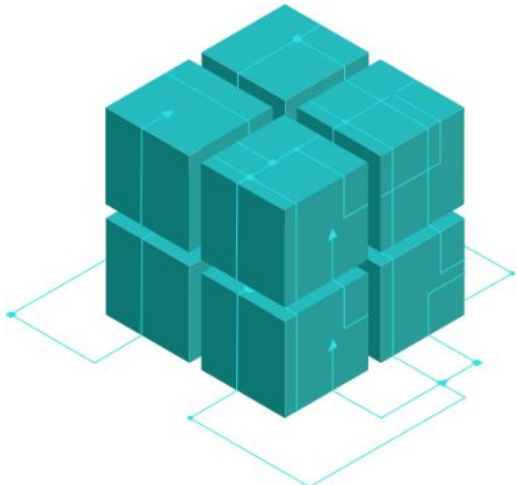
2-3-4 February 2026
Ghent, Belgium

Uyuni

Connecting two distinct worlds of Salt and Ansible

Victor Zhestkov
Software Developer
Uyuni / SUSE Multi-Linux Manager
vzhestkov@suse.com

Pablo Suárez Hernández
Software Developer / Release Engineer
Uyuni / SUSE Multi-Linux Manager
psuarezhernandez@suse.com



Agenda

- Quick introduction to Uyuni
- Uyuni and Ansible journey
  - Ansible integration in Salt
  - Collecting Ansible Inventories
  - Running playbooks
  - Orchestration using Uyuni and Ansible (Action Chains)
- Recent enhancements around Ansible in Uyuni 2026.01
- A next step forward on Ansible integration
  - Running Ansible reusing Salt transport
- Demo

What is Uyuni?

“Salar de Uyuni” is the world's largest salt flat*

Uyuni

/uju:ˈni/

What is Uyuni?

- Scalable systems management solution
- Deploy and manage all kinds of Linux workloads, wherever they are, from a single place
- Hardware and software inventories
- Configuration management: automatically maintain standard configurations
- Granular control over content delivery
- Powered by Salt / OpenJDK / ReactJS
- WebUI / CLI / API
- Now, containerized!

Uyuni features you can use!

- Transparent integration with Salt
- Manage on-prem, cloud, hybrid cloud or multi-cloud systems
- Content Lifecycle Management: define states (DEV, TEST, PROD, ...) for your software channels.
- Apply filters to add/remove contents and create new channels
- Recurring actions, states and highstate. Maintenance window.
- Compliance: CVE audit, SCAP, subscription matching
- Monitoring (Prometheus & Grafana stack), including federation
- Formulas with Forms: create YAML automation templates, no programming skills required!

More cool features

- Image creation
  - Kiwi or Docker
  - Using managed channels
- Autoinstallations (Autoyast / Kickstart profiles)
- Connection to other environments
  - VMware
  - Nutanix AHV
  - Public Clouds
- Confidential Computing
- Ansible Integration: operates your Ansible control node (run playbooks)

More and more Linux distros

EoL

- RHEL 6/7
- CentOS 6/7/8
- Oracle 6/7
- Debian 9/10
- Ubuntu 16.04/18.04
- Debian 11

Active

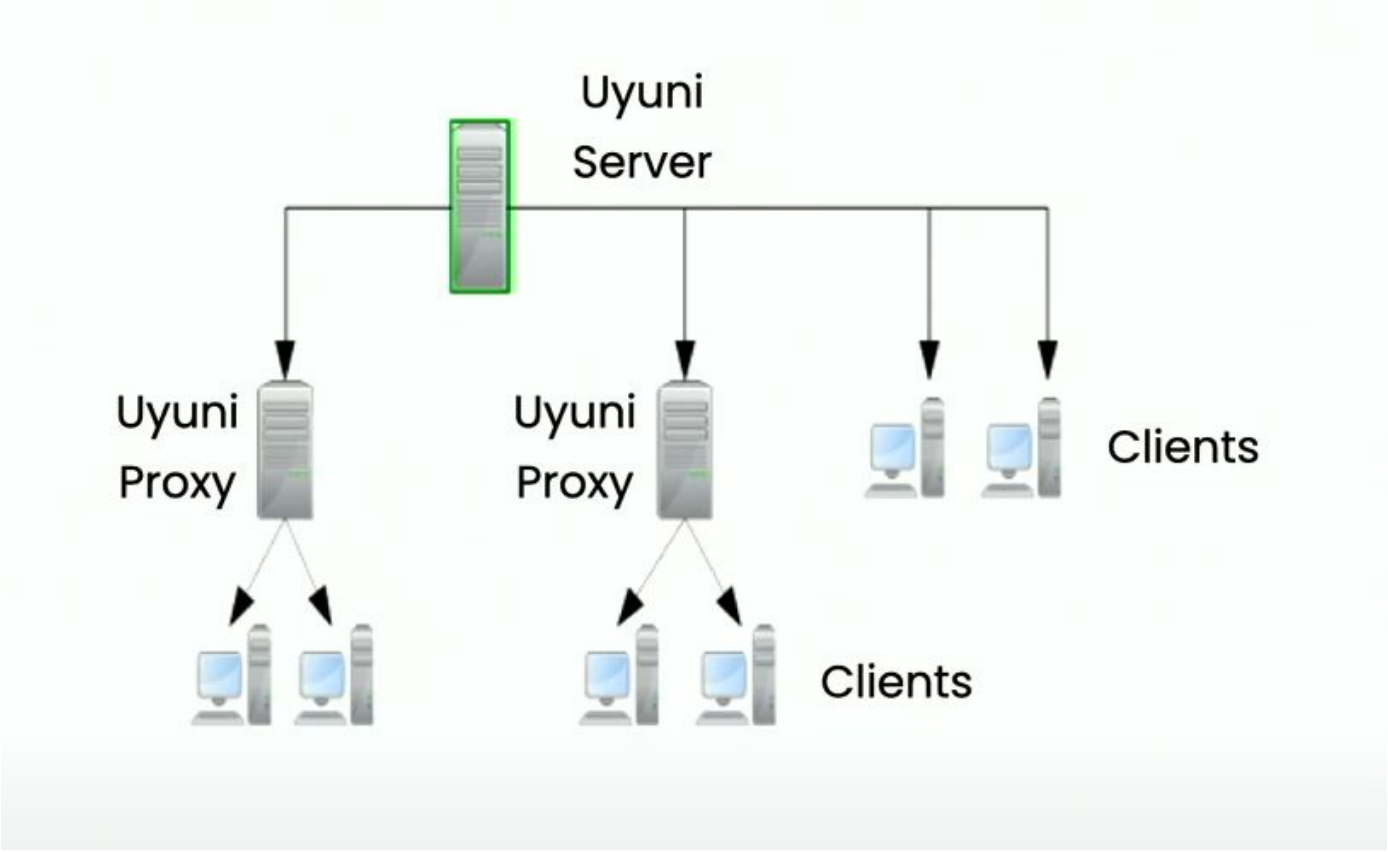
- SUSE Linux Enterprise 12/15
- SUSE Linux Micro 6.X
- openSUSE Leap 15.x
- openSUSE Leap Micro
- Alma Linux 8/9
- Rocky Linux 8
- Rocky Linux 9
- Oracle Linux 8/9
- Debian 12
- Raspberry Pi OS 12
- Ubuntu 20.04
- Ubuntu 22.04
- Ubuntu 24.04
- powered by aws Amazon Linux 2
- powered by aws Amazon Linux 2023
- openEuler 22.03
- Alibaba Linux 2
- RHEL 8/9

Upcoming

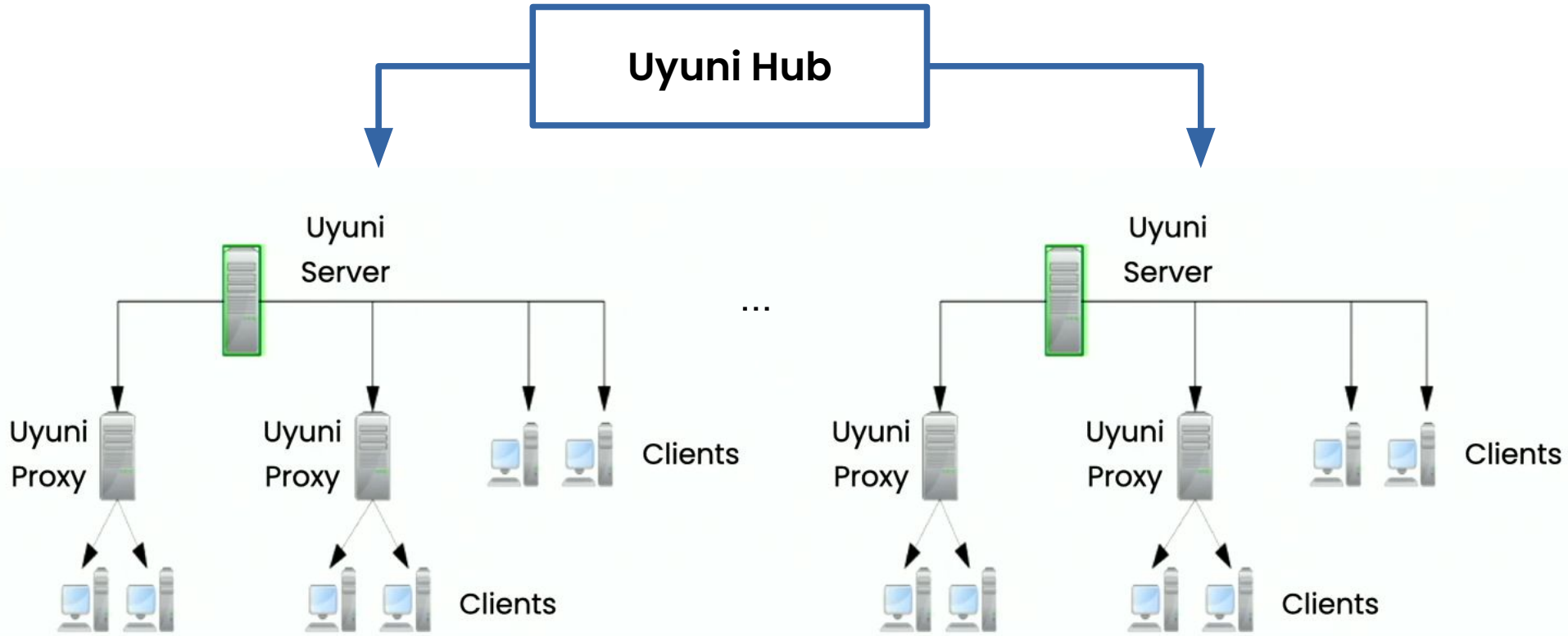
- SUSE Linux Enterprise 16
- openSUSE Leap 16.X
- Zorin OS 17.2
- FUSS 12
- ....?

Contribution by the Uyuni Community
Contribution during SUSE's Hack Week

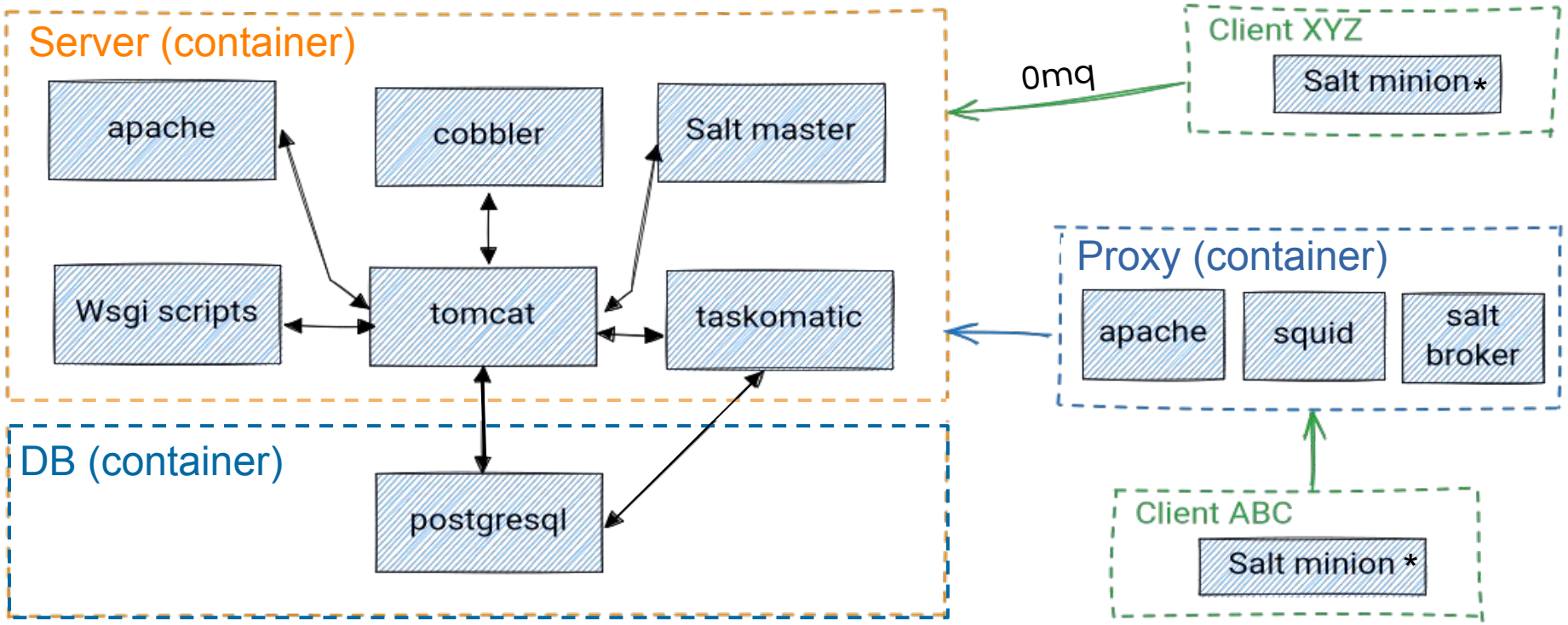
Uyuni Architecture



Uyuni Architecture

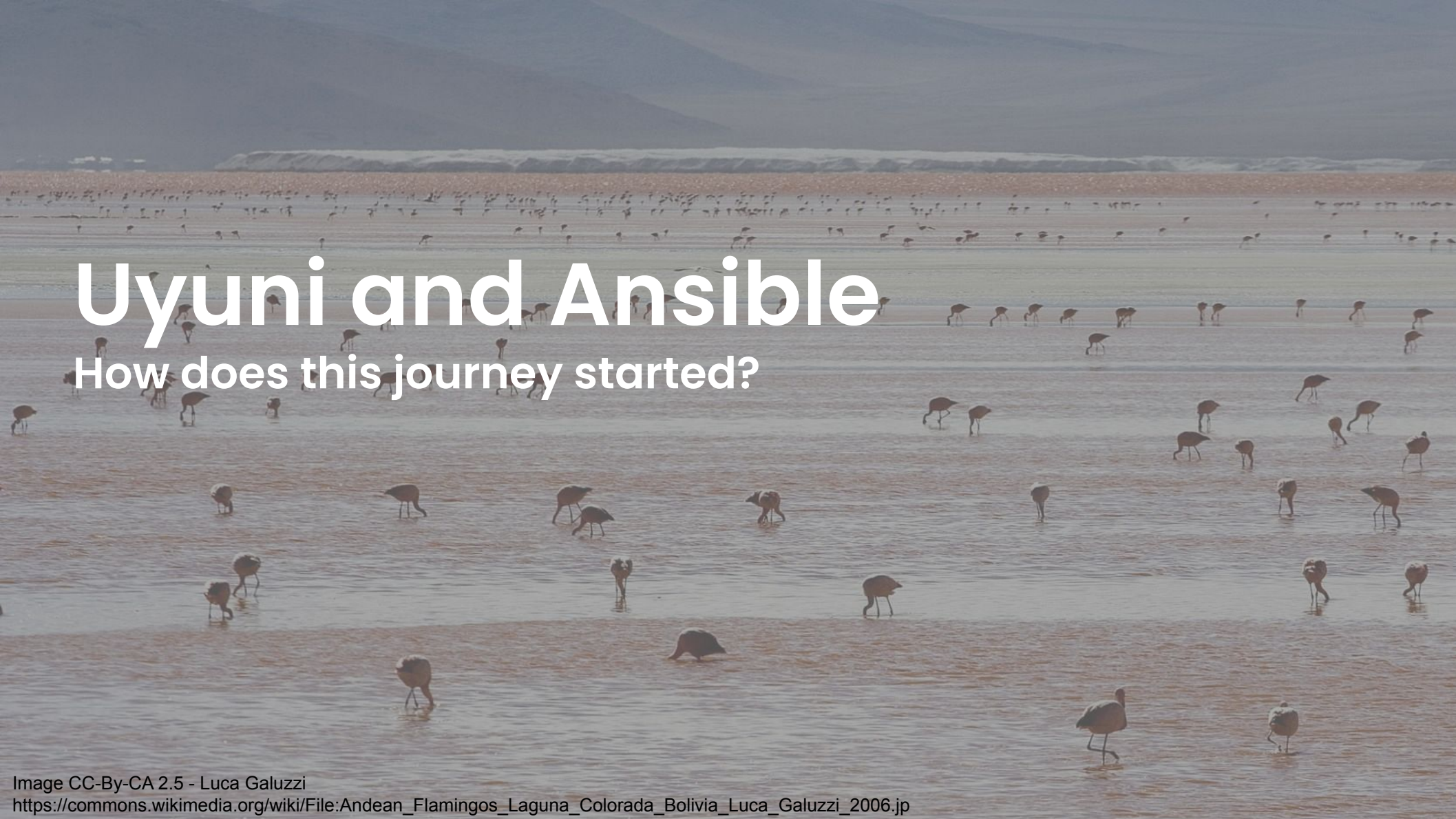


Looking under the hood



* Salt Bundle (aka venv-salt-minion)
* No agent installed if registered as SSH only

So yes… Uyuni is a big beast!



Uyuni and Ansible

How does this journey started?

Uyuni and Ansible

- Uyuni uses Salt natively but …
- There was a necessity: people are using Ansible out there!
- More and more people adopting Ansible
- OK, we need to allow users to use Ansible in Uyuni as well!

Uyuni and Ansible

- Uyuni uses Salt natively but ...
- There was a necessity: people are using Ansible out there!
- More and more people adopting Ansible
- OK, we need to allow users to use Ansible in Uyuni as well!

We need to make Salt and Ansible to become friends :)

Uyuni and Ansible

Around 2018, we introduced the "**ansiblegate**" module for Salt (thanks Bo!)

- Execute the Ansible modules using Salt
- Run your playbooks using Salt

https://docs.saltproject.io/en/3006/ref/modules/all/salt.modules.ansiblegate.html

Uyuni and Ansible

Around 2018, we introduced the “**ansiblegate**” module for Salt (thanks Bo!)

- Execute the Ansible modules using Salt
- Run your playbooks using Salt

```
$ salt ‘minion’ ansible.call ping data=foobar
...
$ salt 'controlnode' ansible.playbooks /srv/playbooks/play.yml
```

Uyuni and Ansible

Around 2018, we introduced the "**ansiblegate**" module for Salt (thanks Bo!)

- Execute Ansible modules with Salt
- Run your playbooks using Salt

```
$ salt 'minion' ansible.call ping data=foobar
...
$ salt 'controlnode' ansible.playbooks /srv/playbooks/play.yml
```

```
$ cat /srv/salt/mytest.sls
my_test_state:
  ansible.playbooks:
    - playbook: /srv/playbooks/foobar.yml
    - extra_vars: {"var1": "foo", "var2": "deadbeaf"}
```

Uyuni and Ansible

Around 2018, we introduced the "**ansiblegate**" module for Salt (thanks Bo!)

- Execute Ansible modules with Salt
- Run your playbooks using Salt

```
$ salt 'minion' ansible.call ping data=foobar
...
$ salt 'controlnode' ansible.playbooks /srv/playbooks/play.yml
```

- Only available for Salt CLI, not yet (2018) integrated with Uyuni UI / API
- Helpful for plain Salt users, but you still need to know about Salt to use it

Uyuni and Ansible

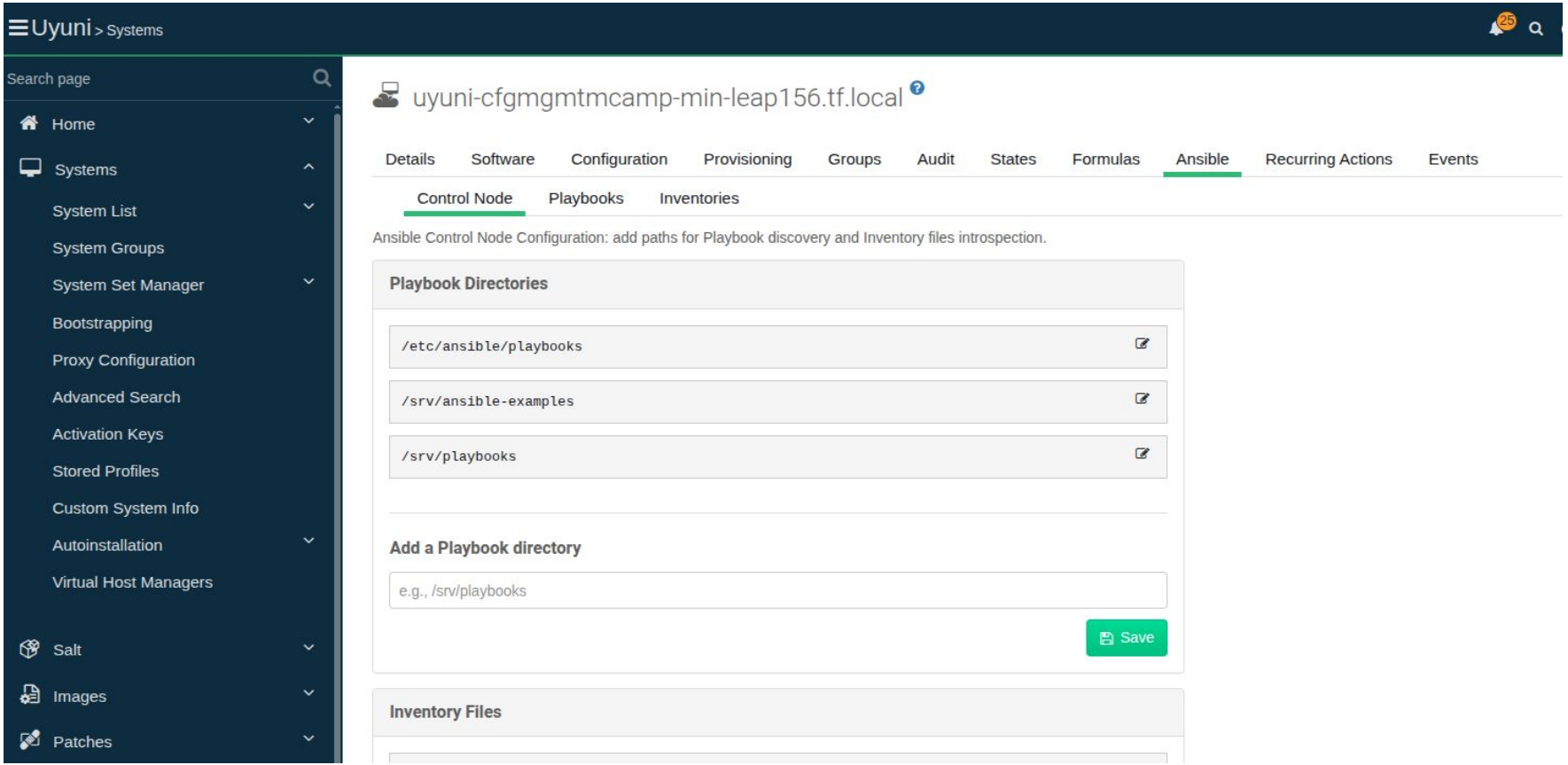
Integration in Uyuni UI / API

Uyuni and Ansible

Initial integration of Ansible in Uyuni in version 2021.06 (tech preview)

- Allow to operate your Ansible control node using Uyuni
- Add your inventories to Uyuni, discover your playbooks
- Trigger playbooks execution using Uyuni UI / API (Salt used under the hood)
- Integrate your playbooks execution with the rest of your Uyuni actions (i.a. Action Chains, Maintenance Windows, Salt highstate, ...)
- Coexistence of Ansible and Salt managed clients

Uyuni and Ansible



Uyuni and Ansible

Recent enhancements in Uyuni 2026.01

- Playbook’s variable definition & editing
- Live detection of changes in your Ansible inventories
- Improved output from playbooks executions

Uyuni and Ansible

Recent enhancements in Uyuni 2026.01

- Playbook's variable definition & editing
- Live detection of changes in your Ansible inventories
- Improved output from playbooks executions
- Still some limitations / potential enhancements:
  - An Ansible control node is still required as a managed client
  - Still a separation between Ansible / Salt managed clients

DEMO 1

Image CC-By-CA 2.5 - Luca Galuzzi
https://commons.wikimedia.org/wiki/File:Andean_Flamingos_Laguna_Colorada_Bolivia_Luca_Galuzzi_2006.jp

Uyuni and Ansible

So, what's coming next?

Focus of Hack Week project

Ansible to Salt integration from the other point of view

- Make Ansible easier to use in an existing Salt environment
- Uyuni/SUSE Multi-Linux Manager are out of focus, only pure Ansible and Salt were used for the PoC project
- Salt-SSH is out of focus (but possible)
- **Main goal**: avoid creating new connections to the managed clients, only use existing Salt/ZeroMQ transport established by the minions to the master
- **Bonus**: make using Ansible in the existing Salt environment seamless and transparent

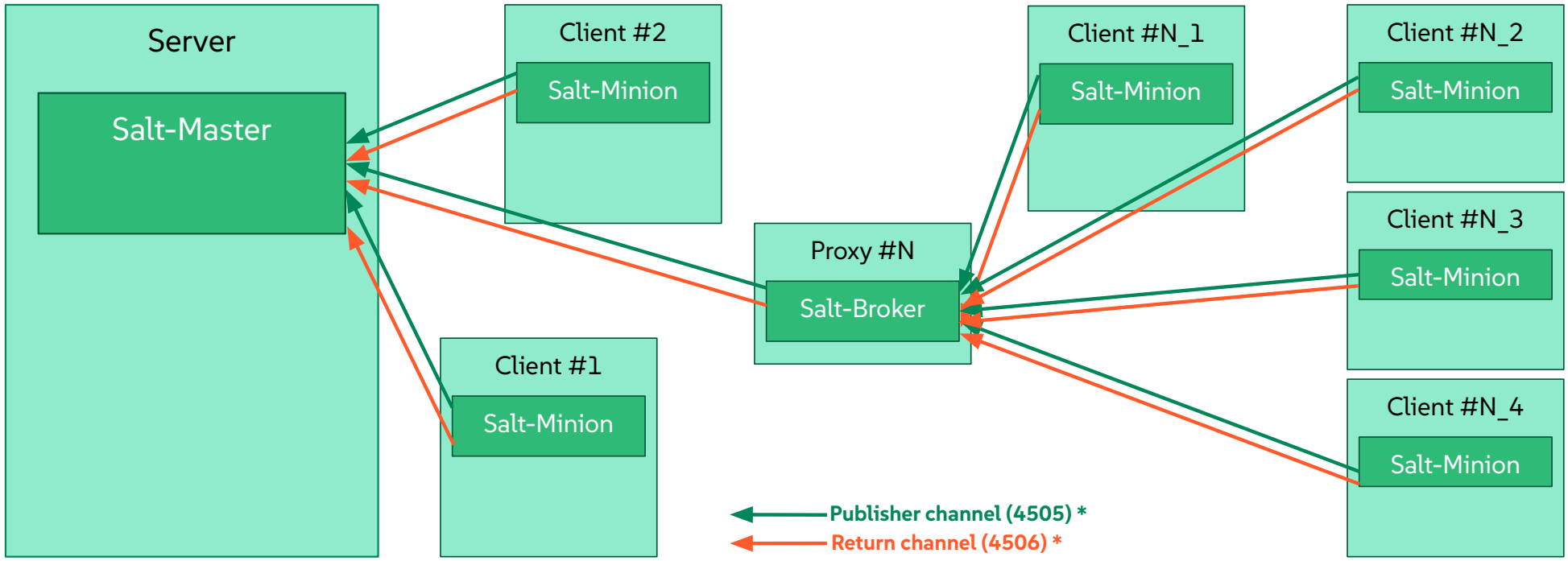
Ansible Integration as AI

We have to use AI in the modern world

Current state of Ansible integration in Uyuni

Salt environment example

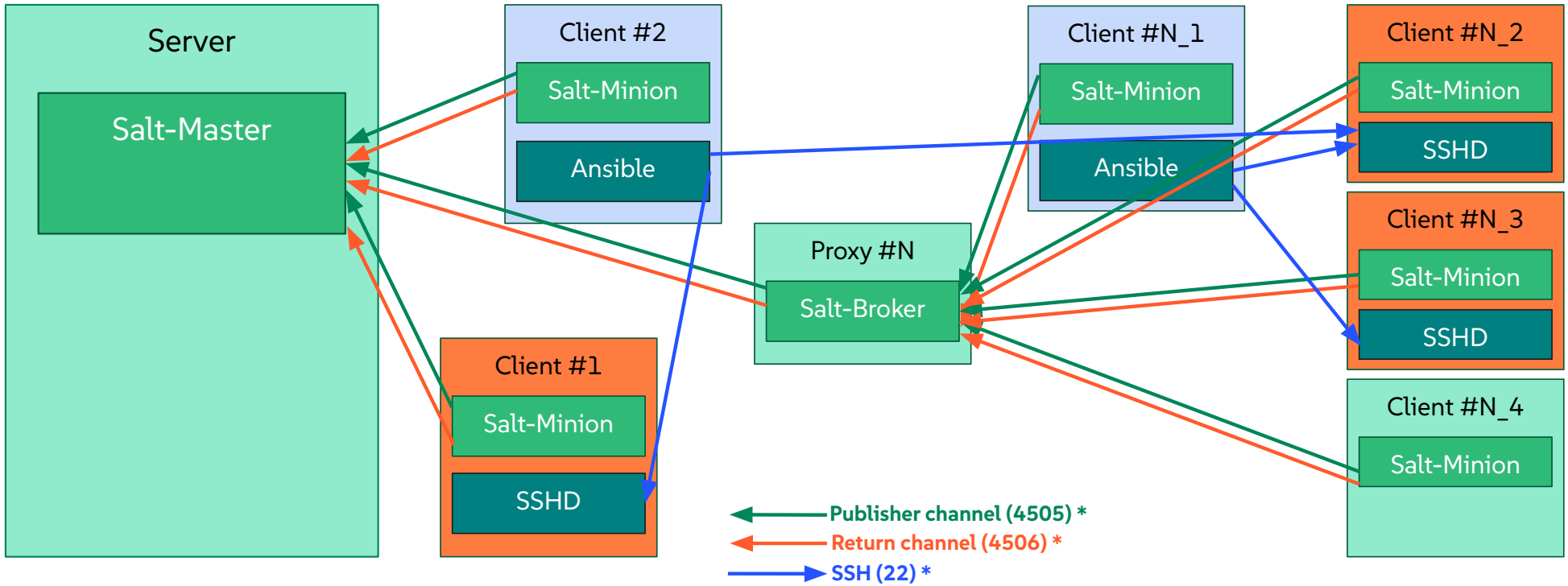
The example of pure salt environment with proxy



* The arrows are representing the direction of connection initiation, NOT the data flow.

Existing Ansible Integration

What is already there in Uyuni/SUSE Multi-Linux Manager

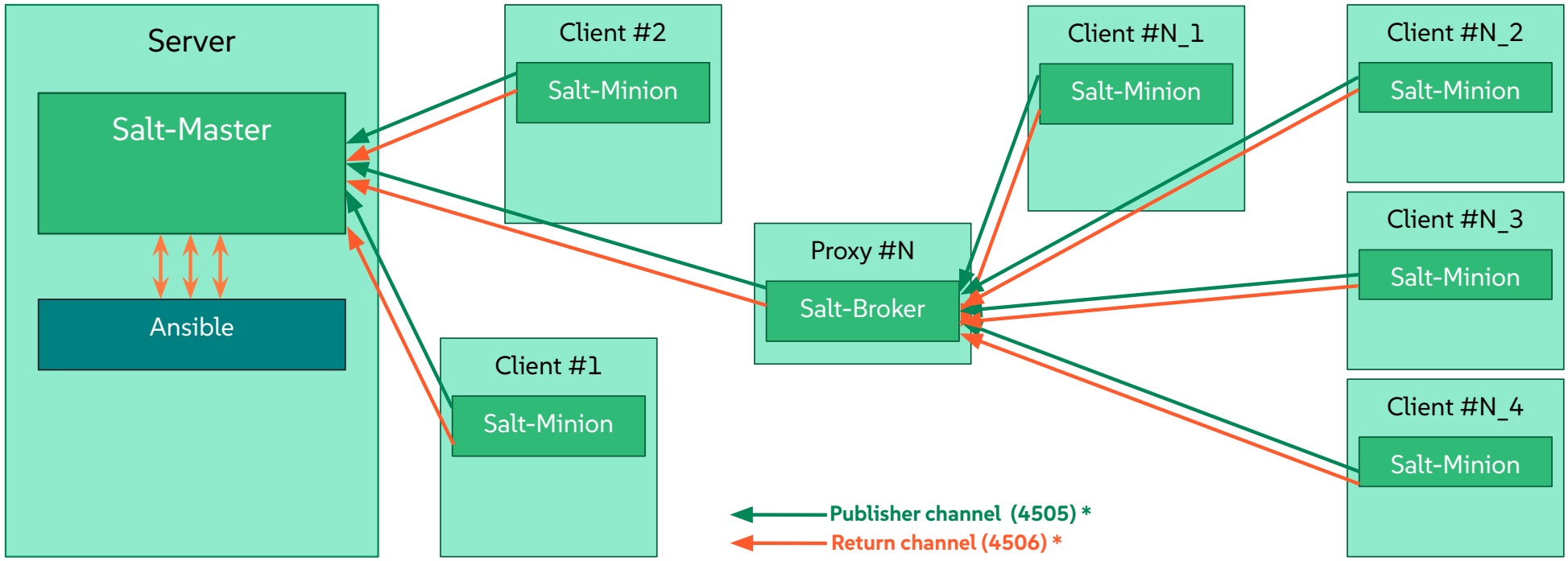


* The arrows are representing the direction of connection initiation, NOT the data flow.

What is the target of this project?

Ansible is using existing Salt/ZeroMQ as a transport

There is no new connections to the clients made while processing Ansible calls



* The arrows are representing the direction of connection initiation, NOT the data flow.

DEMO 2

Image CC-By-CA 2.5 - Luca Galuzzi
https://commons.wikimedia.org/wiki/File:Andean_Flamingos_Laguna_Colorada_Bolivia_Luca_Galuzzi_2006.jp

Results

Image CC-By-CA 2.5 - Luca Galuzzi
https://commons.wikimedia.org/wiki/File:Andean_Flamingos_Laguna_Colorada_Bolivia_Luca_Galuzzi_2006.jp

Results

The outcome of the project

Good things

- **It works!**
- No need to store any credentials
- Fully transparent
- Almost seamless
- Ansible playbooks and collections can be used with no changes
- **Bonus**: it can use Python from Salt Bundle as an Ansible Interpreter

Bad things

- There is an extra workload to the salt-master
- Possible performance issues
- Overall traffic to the minions is higher than to plain SSH with Ansible calls

Unknown things

- Uyuni system groups to Ansible groups translation
- Ownership of the systems assigned to different organizations
- Uyuni users permission assignment

Some measurements to consider

The results of very synthetic test of comparison pure Salt and Ansible and using this integration

	Target client (avg.)	
	IN	**OUT**
salt CLIENT test.ping	1115	1198
ansible -m ping CLIENT		
hashutil.base64_decodefile *	168181	248144
cp.recv_chunked	126378	127763
cp.get_file	131794	16595
cp.get_file (gzip)	102078	18234
-c ssh (native ansible)	91341	5899

Some measurements to consider

The results of very synthetic test of comparison pure Salt and Ansible and using this integration

	Target client (avg.)		Other client (avg.)	
	IN	**OUT**	**IN**	**OUT**
salt CLIENT test.ping	1115	1198	517	52
ansible -m ping CLIENT				
hashutil.base64_decodefile *	168181	248144	163226	936
cp.recv_chunked	126378	127763	120422	941
cp.get_file	131794	16595	5325	416
cp.get_file (gzip)	102078	18234	6544	416
-c ssh (native ansible)	91341	5899	0	0

Join our community

- Contribute with ideas and feedback: mailing list, Gitter, Github
- Uyuni Community Hours: Every last Thursday of the month, 16.00 CET/CEST
  - https://calendar.opensuse.org/teams/uyuni/events/uyuni-community-hours
- Contribute with code:
  - Set up your development environment: https://github.com/uyuni-project/uyuni/wiki
  - Hack & submit a pull request
- Contribute with translations:
  - https://l10n.opensuse.org/projects/uyuni

Q&A

/uyuni-project

/uyuni-project

/UyuniProject

Join Us at uyuni-project.org

License

General Disclaimer

Credits

Template & Design
Pau Garcia Quiles

Inspiration
openSUSE Design Team
http://opensuse.github.io/branding-guidelines/