PERMISSIONS & SECURITY IN PLONE

Kim Nguyen / kim@sixfeetup.com / sixfeetup.com

six feet up

ABOUT ME

- Director of Engineering @ Six Feet Up
- Have worked Python since 1997, Plone since 2003
- Developer → team lead → project manager
- 350+ Plone sites at University of Wisconsin Oshkosh, including the campus Intranet
- Plone Foundation Board (2014-2019), Plone Conference organizer (2016, 2017), Plone Symposium Midwest organizer (2013, 2014)

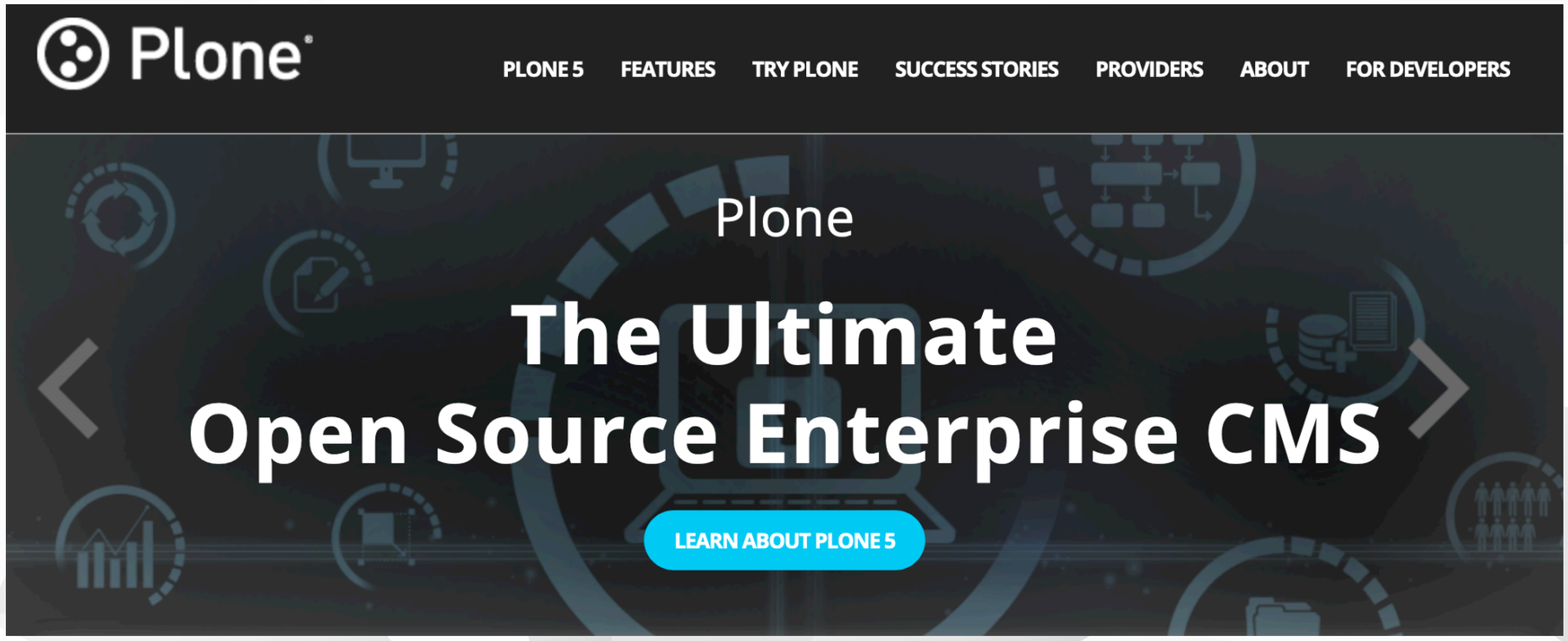
six feet up 

DATA SECURITY

- "Making Headlines for All the Wrong Reasons"
- Security is hard
- Machine learning is now being used to find vulnerabilities
- New systems, new stacks… SHINY!
- New developers

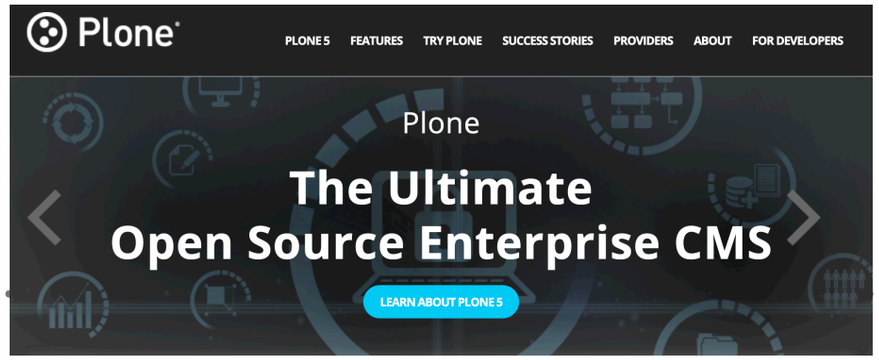
six feet up

PLONE



Secure, Flexible
Content Management In A Box

A HISTORY OF PLONE



- Open source, community-driven since 2001
- Python
- Zope
- Great security track record; no zero days ever
- Experienced security team
- Time tested, robust
- Trusted by governments and large organizations

six feet up

What makes Plone secure?

six feet up

ZOPE

- Baked-in security at the data object level
- Hierarchical model
- ZODB, the Zope Object Database
- Fine-grained permission/role security model

- User - role - permission three layer security model
- Security declarations in ZCML for views, adapters
- RestrictedPython: a subset of Python that can run user-entered programs safely

six feet up

RESTRICTEDPYTHON

- Hardening is done at the Abstract Syntax Tree level
- When you try to add Python code or customize page templates through the Management Interface:
  - scripts and evaluations are compiled specially
  - have limited Python functionality
- Zope's AccessControl & zope.security check every function call against the security manager

six feet up

FIELD-LEVEL PERMISSIONS

- Dexterity content type framework
- All fields default to the same security settings as their containing object
- But you can override them

```
21
22  class ISponsor(model.Schema):
23      """Dexterity Schema for Sponsors
24      """
25
26      directives.widget(level=RadioFieldWidget)
27      level = schema.Choice(
28          title=_(u'Sponsoring Level'),
29          vocabulary=LevelVocabulary,
30          required=True
31      )
32
33      text = RichText(
34          title=_(u'Text'),
35          required=False
36      )
37
38      url = schema.URI(
39          title=_(u'Link'),
40          required=False
41      )
42
43      fieldset('Images', fields=['logo', 'advertisement'])
44      logo = namedfile.NamedBlobImage(
45          title=_(u'Logo'),
46          required=False,
47      )
48
49      advertisement = namedfile.NamedBlobImage(
50          title=_(u'Advertisement (Gold-sponsors and above)'),
51          required=False,
52      )
53
54      directives.read_permission(notes='cmf.ManagePortal')
55      directives.write_permission(notes='cmf.ManagePortal')
56      notes = RichText(
57          title=_(u'Secret Notes (only for site-admins)'),
58          required=False
59      )
```

PERMISSIONS AND ROLES

- Permissions control whether logged-in or anonymous users can execute code and access content
- Roles are sets of permissions
- Users and groups can be assigned roles
- Plone: ~180 permissions, 10 roles

Permissions		Roles										
Acquire?	Permission	Anonymous	Authentic…	Contributor	Editor	Manager	Member	Owner	Reader	Reviewer	Site Admi…	
☑	Access Transient Objects	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
☑	Access arbitrary user session data	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	
☑	Access contents information	☐	☐	☑	☑	☐	☐	☐	☑	☐	☑	
☑	Access inactive portal content	☐	☐	☐	☐	☐	☐	☑	☐	☐	☑	

VIEWS

- Security applies to views too
- e.g.“Site Setup” control panels list

```
<configure
    xmlns="http://namespaces.zope.org/zope">

  <permission id="plone.app.controlpanel.Overview"
              title="Plone Site Setup: Overview">
    <role name="Manager"/>
    <role name="Site Administrator"/>
  </permission>

  <permission id="plone.app.controlpanel.Editing"
              title="Plone Site Setup: Editing">
    <role name="Manager"/>
    <role name="Site Administrator"/>
  </permission>
```

- Permission checks are done for:
  - every view/method which is hit by incoming HTTP request (Plone automatically publishes traversable methods);
  - every called method for RestrictedPython scripts

USERS AND GROUPS

- Permissions and roles can be assigned to a single user
- But… for much better scaling, Plone makes it easy to assign roles to groups
- Pluggable Authentication System lets you connect Plone to central auth systems such as LDAP, OAuth, CAS

six feet up

Wave Robotics
K-12 STEM Programs

WELCOME

Wave Robotics (Team 2826) is the Oshkosh, Wisconsin, USA, team participating in the First Robotics Championship since 2008

SUPPORT WAVE ROBOTICS

See how you can support Team 2826 and its efforts to introduce science, technology, engineering and math to students at schools in Winnebago county

MEMBERSHIP FORMS

Signup forms for the 2019-2020 season are online, for new or returning students, mentors, and volunteers.

RESOURCES AND FORMS FOR CURRENT MEMBERS

Looking for documents and forms? You can find them in our For Members section.

Wave Robotics was started in 2008 with the goal of providing STEM based opportunities to students through the FIRST Robotics program. A goal of Wave Robotics is to retain students in the local area as they enter employment.

Since then, Wave Robotics has expanded to offer K-12 programming in FIRST programs, camps, and different outreach and demonstration initiatives.

There are over 150 students and 40 volunteers engaged in Wave Robotics program offerings each year with thousands more additional students impacted with our outreach and Lego League competitions.

The program has steadily expanded to offer new opportunities, and we are always looking for volunteers. Please visit our contact page if you are interested in volunteering!

News

- Oshkosh FIRST -- First Lego League Update
  Nov 20, 2019
- Wave Competes in the 2019 China Robotics Challenge!
  Nov 13, 2019
- Robotics team earns another trip to Detroit championship
  Mar 27, 2019
- More than a machine: Oshkosh's FIRST Wave students grow alongside their robot
  Apr 24, 2018
- Oshkosh Lego robotics team heads to state
  Feb 23, 2018

More news...

Blog Posts

Week 2 a lesson in patience

Discussion, Reflection, and more Discussions

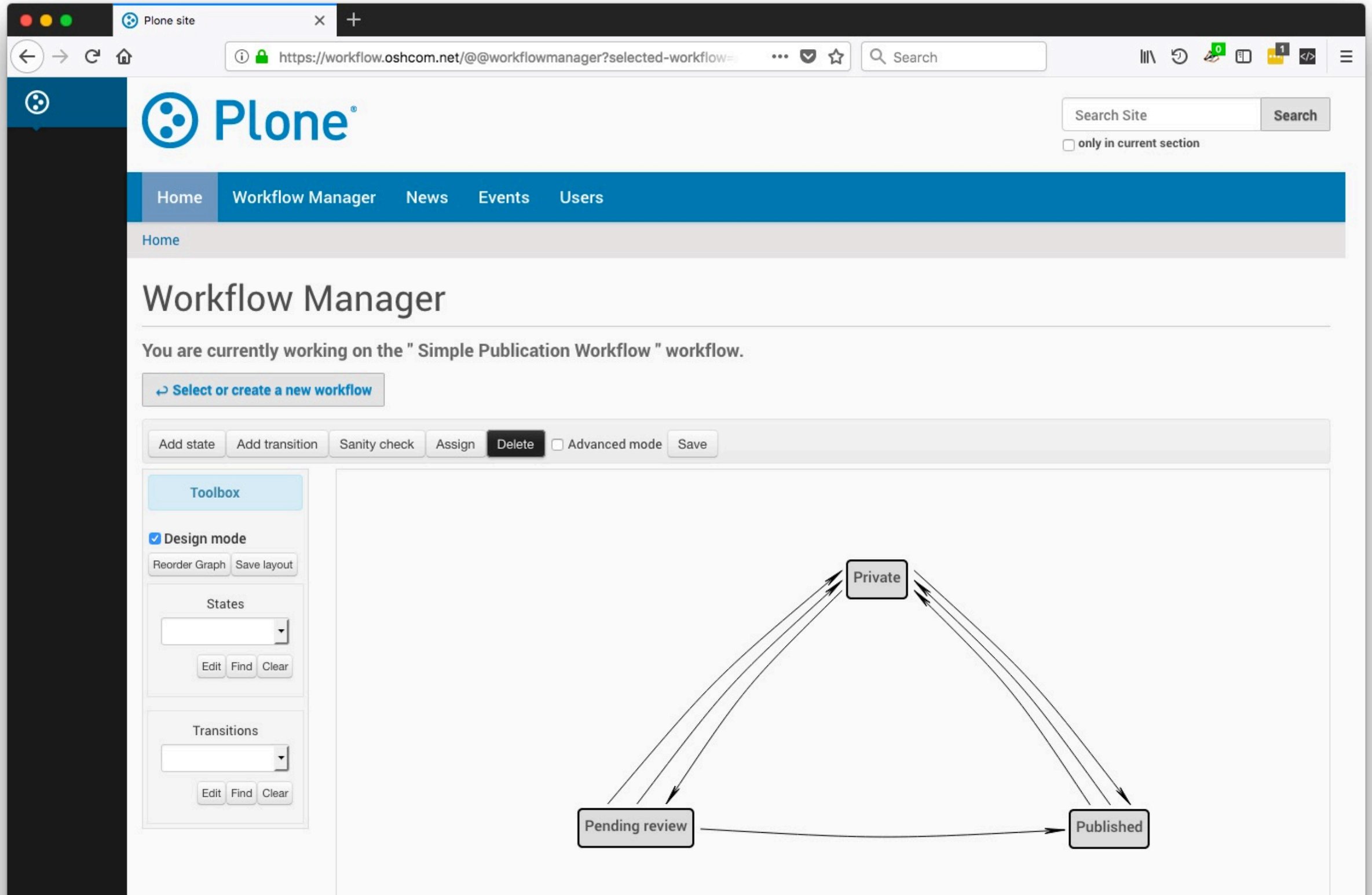
Blog
Programs
Sponsors
About Us

PLONE WORKFLOW

- A conceptually simple way to declare and enforce the security on an object:
  - States and transitions
  - Permissions and roles affected per state
  - Guards on each transition

six feet up

SIMPLE PUBLICATION WORKFLOW



STATE PERMISSION/ROLE MAP

ZOPE 4 | manager | Control Panel | Select type to add

Properties | **Permissions** | Groups | Variables

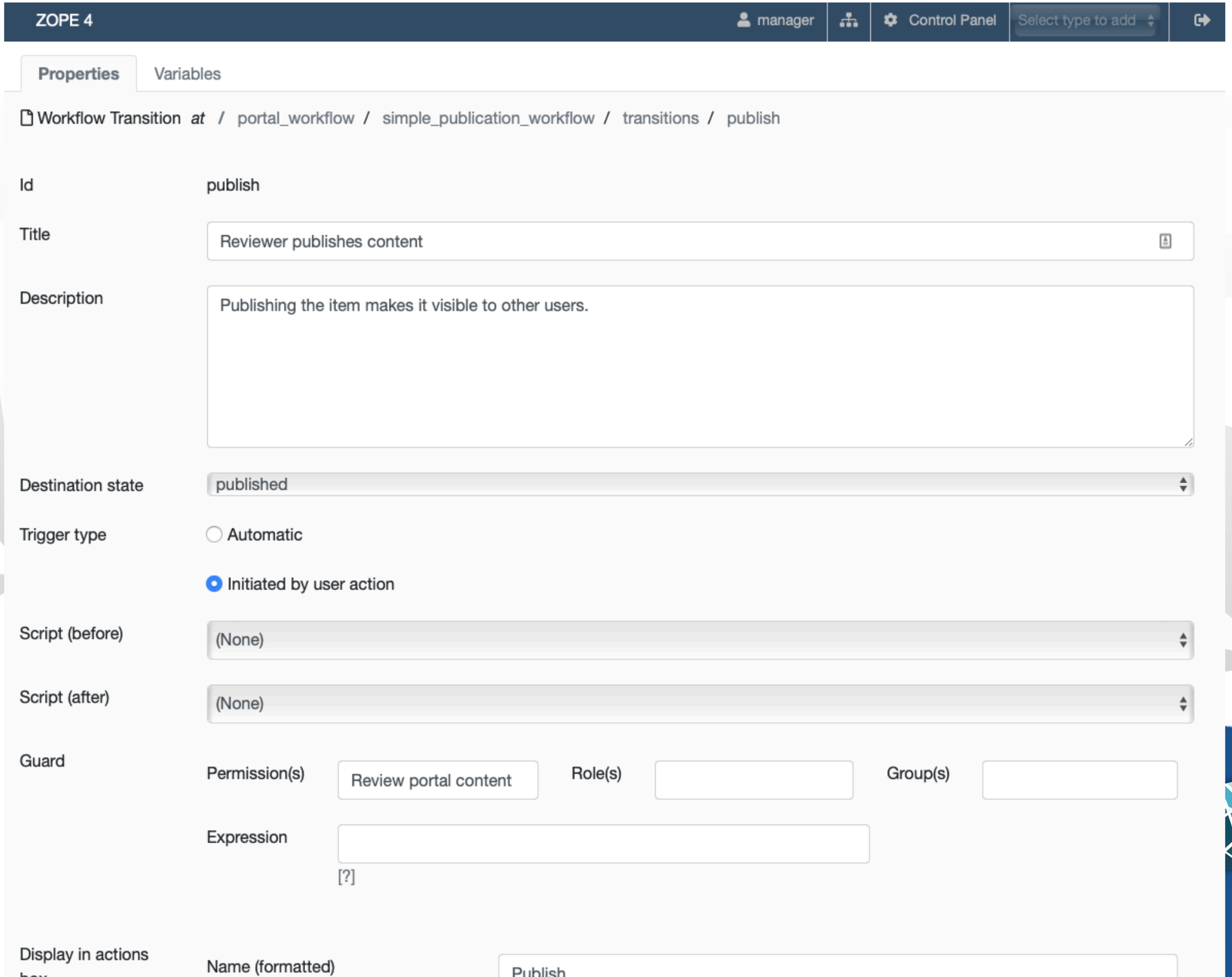
Workflow State *at* / portal_workflow / simple_publication_workflow / states / pending

When objects are in this state they will take on the role to permission mappings defined below. Only the permissions managed by this workflow are shown.

Acquire permission settings?	**Permission**	**Roles**	Anonymous	Authenticated	Contributor	Editor	Manager	Member	Owner	Reader	Reviewer	Site Administrator
☐	Access contents information		☐	☐	☑	☑	☑	☐	☑	☑	☑	☑
☐	Modify portal content		☐	☐	☐	☐	☑	☐	☐	☐	☑	☑
☐	View		☐	☐	☑	☑	☑	☐	☑	☑	☑	☑

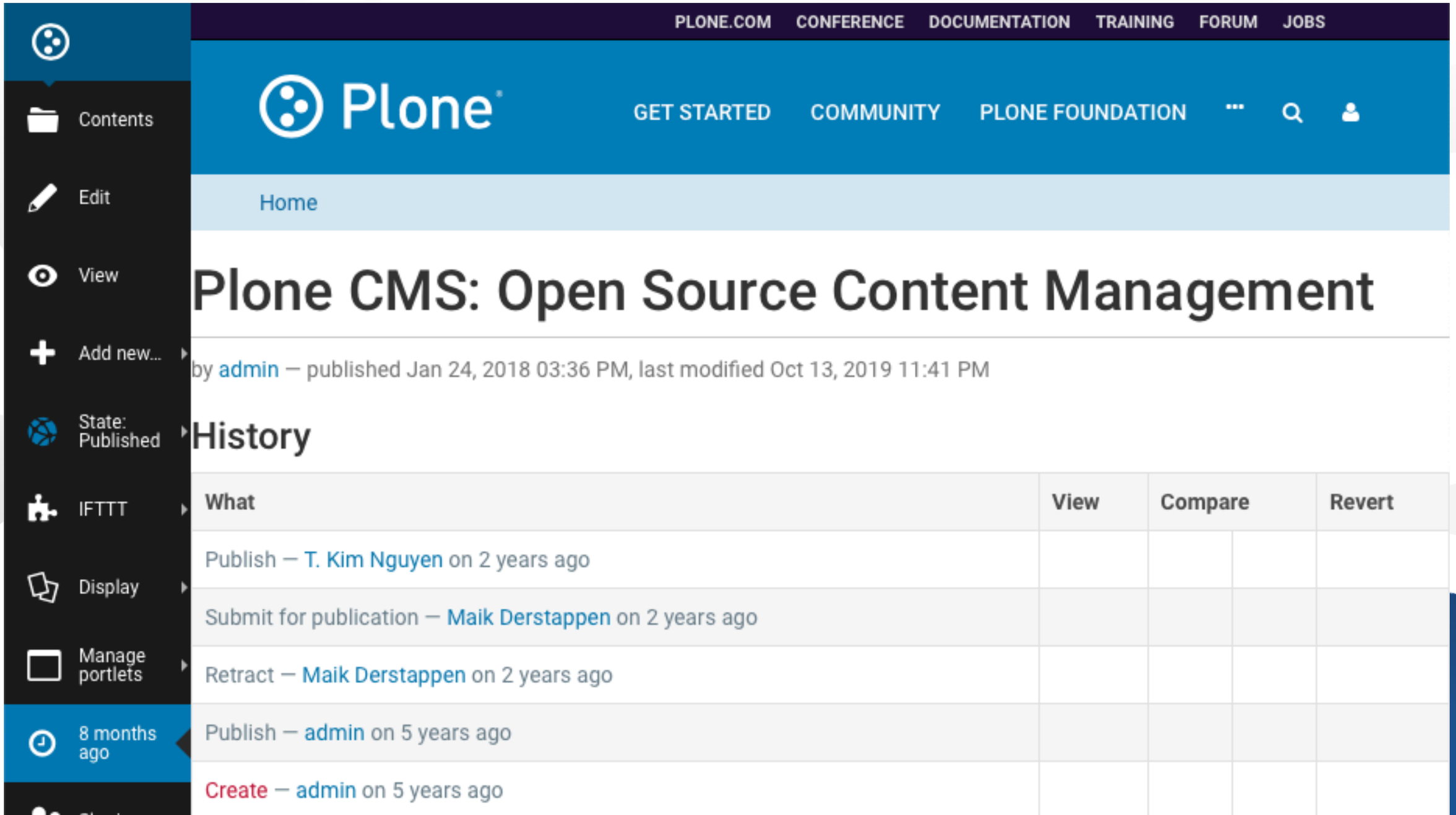
six feet up

TRANSITION DETAILS



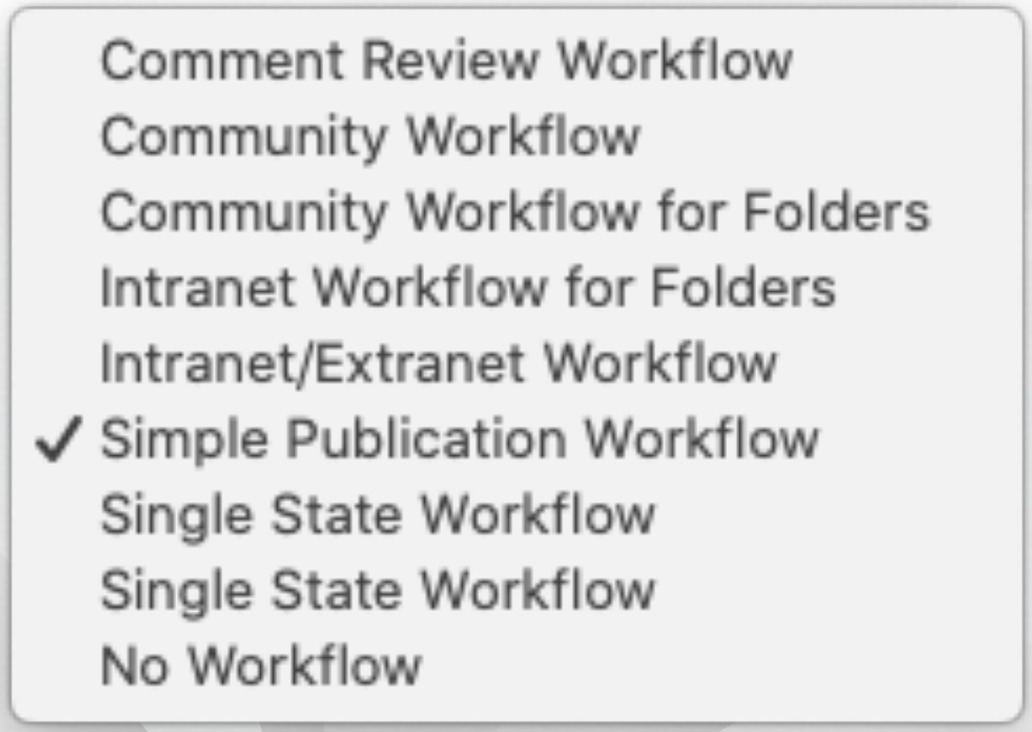
AUDIT TRAIL

- Every workflow state change, or transition, is tracked automatically



OTHER PLONE WORKFLOWS

- Plone ships with several workflows
- You can create your own



six feet up

PROTECTION FROM MALICIOUS CONTENT

HTML Filtering Settings

Keep in mind that editors like TinyMCE might have additional filters.

☐ **Disable HTML filtering** *Warning: disabling this can be dangerous. Only disable if you know what you are doing.*

Nasty tags *These tags and their content are completely blocked when a page is saved or rendered. They are only deleted if they are not marked as valid_tags*

style
object
embed
applet
script
meta

Valid tags *A list of valid tags which will be not filtered out.*

a
abbr
acronym
address
article
aside

Custom attributes *These attributes are additionally allowed.*

six feet up

PLONE.PROTECT

- Helps protect parts of Plone or applications build on top of Plone
- Restricting to HTTP POST
- Form authentication (Cross Site Request Forgery,"CSRF")
- Automatic CSRF protection
  - automatically including the auth token to all internal forms when the user requesting the page is logged in
  - checks for the existence of a correct auth token whenever a request attempts to write to the ZODB
- Clickjacking protection

six feet up

“

What happens when someone browses a Plone site?

A THOROUGH PAT-DOWN

- A comprehensive check of what the user is authorized to do on every requested object, every contained object, every traversed view and method
- If the user is logged in, the check includes:
  - the roles assigned to the user
  - the groups the user is a member of
  - the roles assigned to those groups
  - the permissions assigned to all those roles
  - the permissions that are required for viewing the object and fields
  - the workflows that have been assigned to that object
  - the object's state in each assigned workflow
  - the permission/role map for each state

six feet up

A THOROUGH PAT-DOWN

- If the user is not logged in, it's a bit faster
- If the user is an admin (Manager role) the permission check is cut short, and is very fast

six feet up

THE BEST WAY TO DESECURE PLONE

- Publish everything in your site
- Disable all HTML filtering
- Grant Manager role to every user

Disable HTML filtering

User name	Contributor	Editor	Member	Reader	Reviewer	Site Administrator	Manager
Editor – *editor*	☑	☑	☑	☑	☑	☑	☑
Editor-in-chief – *editorinchief*	☑	☑	☑	☑	⊛	☑	☑
Joe Smith – *joe*	☑	☑	☑	☑	⊛	⊛	⊛
Manager – *manager*	☑	☑	☑	☑	☑	☑	⊛

New workflow: Single State Workflow

- Essentially a workflow with no transitions, but has a Published state, so portlets and applications that expect that state will continue to work.

State Mapping

When changing workflows, you have to select a state equivalent in the new workflow.

Old State	New State
Pending review	Published
Private	Published
Published	Published

six feet up

Alexander Loechel

Actually you could completely deactivate Restricted Python in the Zope Config and thereby deactivate all Security in Zope and Plone, the whole Catalog and Permission System will be disabled

Security Team member

Alexander Loechel

but that needs file system access

six feet up

PLONE: SECURE BY DEFAULT

- Python ✅
- Zope ✅
- ZODB ✅
- Fine-grained permission/role maps ✅
- Users and groups simplify access management ✅
- Object and field-level permissions ✅
- Workflows encapsulate stateful security definitions ✅

six feet up

“

OK, but Plone is old

six feet up

PLONE APIS

- plone.api
  - First release 2012
  - Plone strategic summit (Bristol, UK, 2014)
- plone.restapi
  - First release 2016

six feet up

0.1a1 (2012-07-13)

- Initial release [davisagli, fulv, iElectric, jcerjak, jonstahl, kcleong, mauritsvanrees, wamdam, witsch, zupo]

David Glick ●

Release Engineer at Salesforce.org

😎 Here to hang out

six feet up

PLONE 6

Volto: React-based Front End

THE DUALITY OF PLONE 6

- Rock-solid, time-tested content management system
- And with a modern Volto frontend with React's wide adoption, large community, and great tooling
- Is also a stable, logicful backend store for web and mobile apps (aka headless CMS)

"Icon made by Pixel perfect from www.flaticon.com"

six feet up

NEXT STEPS

- Learn more at plone.com, plone.org
- Get great training materials at training.plone.org
- Volto demo: volto.kitconcept.com, docs.voltocms.com, training.plone.org/5/react
- Volto presentations to watch: 2018.ploneconf.org/talks/plone-react, 2019.ploneconf.org/people/timo-stollenwerk, 2019.ploneconf.org/people/victor-fernandez-de-alba
- Join the forum community.plone.org, attend the annual ploneconf.org

six feet up

six feet up

python & cloud expert consulting

- Planning
- Development
- Orchestration
- Support

QUESTIONS? PLEASE ASK!

KIM@SIXFEETUP.COM

SIXFEETUP.COM