

## Quick Checklist

Django application

uwsgi installation (server admin) <https://uwsgi-docs.readthedocs.org/en/latest/>

deploying base overarching UI

Installing UVCDAT (Install Team)

installing and deploying CDATWEB

Installing ESGF node stack (compute, idp, index, data)

Install and deploy CoG ESGF web-fe (CoG team)

Create page templates

map out and service api

interface with CDATWEB's (API?)

interface with ESGF / CoG / Publisher

interface with AKUNA (example's in repo)

## Todo / Tasks

### I. Infrastructure

Stand up acme-ui

work with john Q to install and stand up acme-ui as it was working at the demo

install all required packages (working on list)

install apache and uwsgi to server the app

test app is reachable from out side

add database for balding user info and model interaction see #2

acme-ui db schema creation --> ORM creation

users

groups

models

alerts

users have alerts

users have models

users have groups

model have comments

model have likes

users have comments

users have likes

Stand up acme-cdatweb

work with john Q and Jon B to install and stand up cdatweb as it was working at the demo

jon b would fill these out

Stand up acme-viewer

work with john Q and John H to install and stand up diagnostics/  
classic view as it was working at the demo

john h would fill these out

Stand up acme-test -> alfresco

work with Akuna to make alfresco work again

test with acme-ui that interaction api is working

Stand up acme-publish / acme-esgf / acme-cog ?

work with John Q and publishing team to stand up VM  
publishing team would install there app and explain how it works to us

## II. ACME-UI Service layer

User log in currently with Globus online

add in GO token / user sessions

Change out login for Django based user auth

Receive User\_id & Model\_id from Viewer

add url to urls.py

read input and pass to correct function in views.py

return a page for seamless app transfer

Receive User\_id & Model\_id from CDATWeb

add url to urls.py

read input and pass to correct function in views.py

return a page for seamless app transfer

Receive User\_id & Model\_id from Publisher

add url to urls.py

read input and pass to correct function in views.py

return a page for seamless app transfer

## III. Model Output connections

Pass Viewer User\_id & Model\_id

from display.html post data to views.py

issues redirect using service call from Viewer

Pass CDATWeb User\_id & Model\_id

same

Pass CDATWeb User\_id & Model\_id & ARGS\* to Publisher

same

## IV. User interactions

comment on any model run

share model with another user

set permissions on model

1 - 3 save and display data

compair model run X with Y

NO CLUE

create missing pages / user profile / user space / random jumping page  
| links / login

## Todo / Tasks

### I. Infrastructure

Stand up acme-ui

work with john Q to install and stand up acme-ui as it was working at  
the demo

install all required packages (working on list)

install apache and uwsgi to server the app

test app is reachable from out side

add database for balding user info and model interaction see #2

acme-ui db schema creation --> ORM creation

users

groups

models

alerts  
users have alerts  
users have models  
users have groups  
model have comments  
model have likes  
users have comments  
users have likes  
Stand up acme-cdatweb  
work with john Q and Jon B to install and stand up cdatweb as it was  
working at the demo  
jon b would fill these out  
Stand up acme-viewer  
work with john Q and John H to install and stand up diagnostics/  
classic view as it was working at the demo  
john h would fill these out  
Stand up acme-test -> alfresco  
work with Akuna to make alfresco work again  
test with acme-ui that interaction api is working  
Stand up acme-publish / acme-esgf / acme-cog ?  
work with John Q and publishing team to stand up VM  
publishing team would install there app and explain how it works to us  
II. ACME-UI Service layer  
User log in currently with Globus online  
add in GO token / user sessions  
Change out login for Django based user auth  
Receive User\_id & Model\_id from Viewer  
add url to urls.py  
read input and pass to correct function in views.py  
return a page for seamless app transfer  
Receive User\_id & Model\_id from CDATWeb  
add url to urls.py  
read input and pass to correct function in views.py  
return a page for seamless app transfer  
Receive User\_id & Model\_id from Publisher  
add url to urls.py  
read input and pass to correct function in views.py  
return a page for seamless app transfer  
  
III. Model Output connections  
Pass Viewer User\_id & Model\_id  
from display.html post data to views.py  
issues redirect using service call from Viewer  
Pass CDATWeb User\_id & Model\_id  
same  
Pass CDATWeb User\_id & Model\_id & ARGS\* to Publisher  
same  
IV. User interactions  
comment on any model run  
share model with another user

set permissions on model  
1 - 3 save and display data  
compare model run X with Y  
NO CLUE  
create missing pages / user profile / user space / random jumping page  
| links / login