

Bram van der Werf London, April 5, 20101

# **OPF Practitioners & Developers Community**

- Stronger focus on requirements
- Priorities & Scoping
- Sharing vs re-inventing
- Deployment & engineering
  - http://openplanetsfoundation.org
  - http://wiki.opf-labs.org

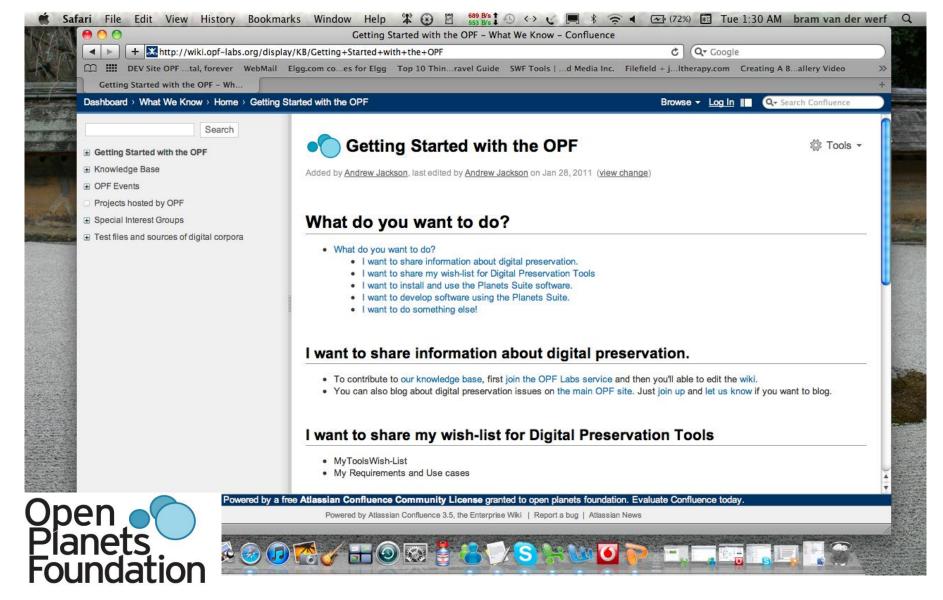


#### **OPENPLANETSFOUNDATION.ORG**

- Practitioner blogs (openplanets.org)
- Preservation WIKI (wiki.opf-labs.org)
- Bug, Issue and feature tracking (jira.opflabs.org/secure/Dashboard.jspa)
- Open Source Development Collab (
- Preservation Community creating requirements and use cases



### Check it out @



### **OPF Developers Community**

- Collaboratory
  - Focus on practitioner requirements
  - Make sure we don't re-invent the wheel)
  - Coding practices
  - Maintainability, Serviceability, Robustness
  - Transparent (Open Source)
  - Keep it simple



### **Managing Open Source Software**

- Github (in sync with SVN on Sourceforge)
- JIRA (Issue Tracking)
- Confluence (Wiki)
- Bamboo (continuous integration)
- Crucible (code review)
- Crowd (single sign-on)
- Fisheye (notification of code changes)



### The challenge of tools and services

- They need ACTIVE Users
- Simple and fit for its purpose
- Supportable and maintainable
- Able to adapt to changing requirements
  - Support and Maintenance



### **Products & Services Roadmap**

Planning

Testbed

Pricing

Fido

Test corpora

Fornat registry editing tool

OS collaboratory

Practitioners WIKI

Developers WIKI

Q1 2011	Q2 2011	Q3 2011	Q4 2011
prototype	beta	beta	supported
prototype	beta	beta	supported
prototype	prototype	prototype	beta
prototype	prototype	beta	beta
not available	available	available	available
prototype	prototype	beta	beta
available	available	available	available
available	available	available	available
available	available	available	available



### Analoge vs Digital

- Usage decays quality of analoge information and has negative impact on longevity, dark archives and limited access
- Usage has no negative impact on quality of digital information, it actually assures longevity. It can trigger a timely response to decay



## Users, Access & Curation, Maintenance

Storage, bit rot

Rendering and Software legacy

Formats and Obsolescence



## Usage as another key to Longevity

- Active Users
  - Are aware of what they need
- Reporting errors
  - Realize when things go wrong
- Feature requests
  - Know what would make things better

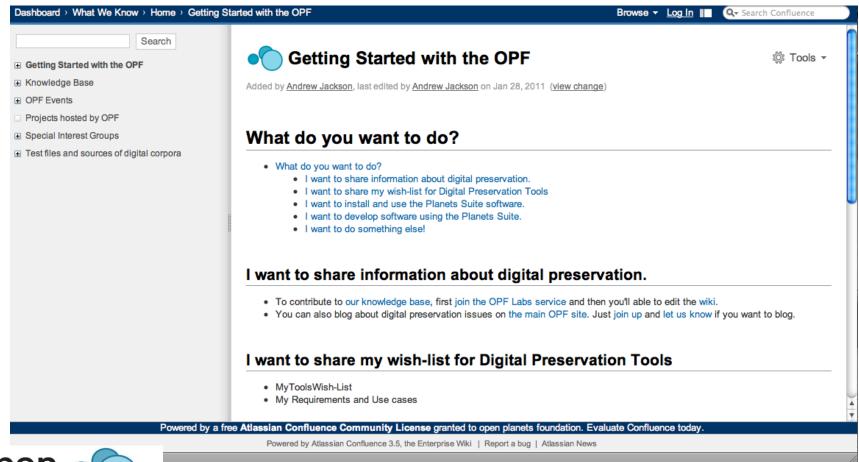


## Longevity Time Machine

- User Community
- Software Maintenance
- Application Management
- Continuous R&D
- Sustainable infrastructure



### Requirements





### From R&D to Service

- Requirements
- Prototypes
- Engineering
- Maintenance

• Users, Users, Users.....



### Deployability

- Multiple platforms, JRE
- Small apps that add functionality to existing middleware
- Engineer R&D prototypes to supported beta (best effort)



### Summary

- Requirements (facilitate)
- Engineering (support and guidance)

Support tools that are actively used