Asset Manager

Rémi Arnaud
David Gaya
Maxime Helen

Asset Manager
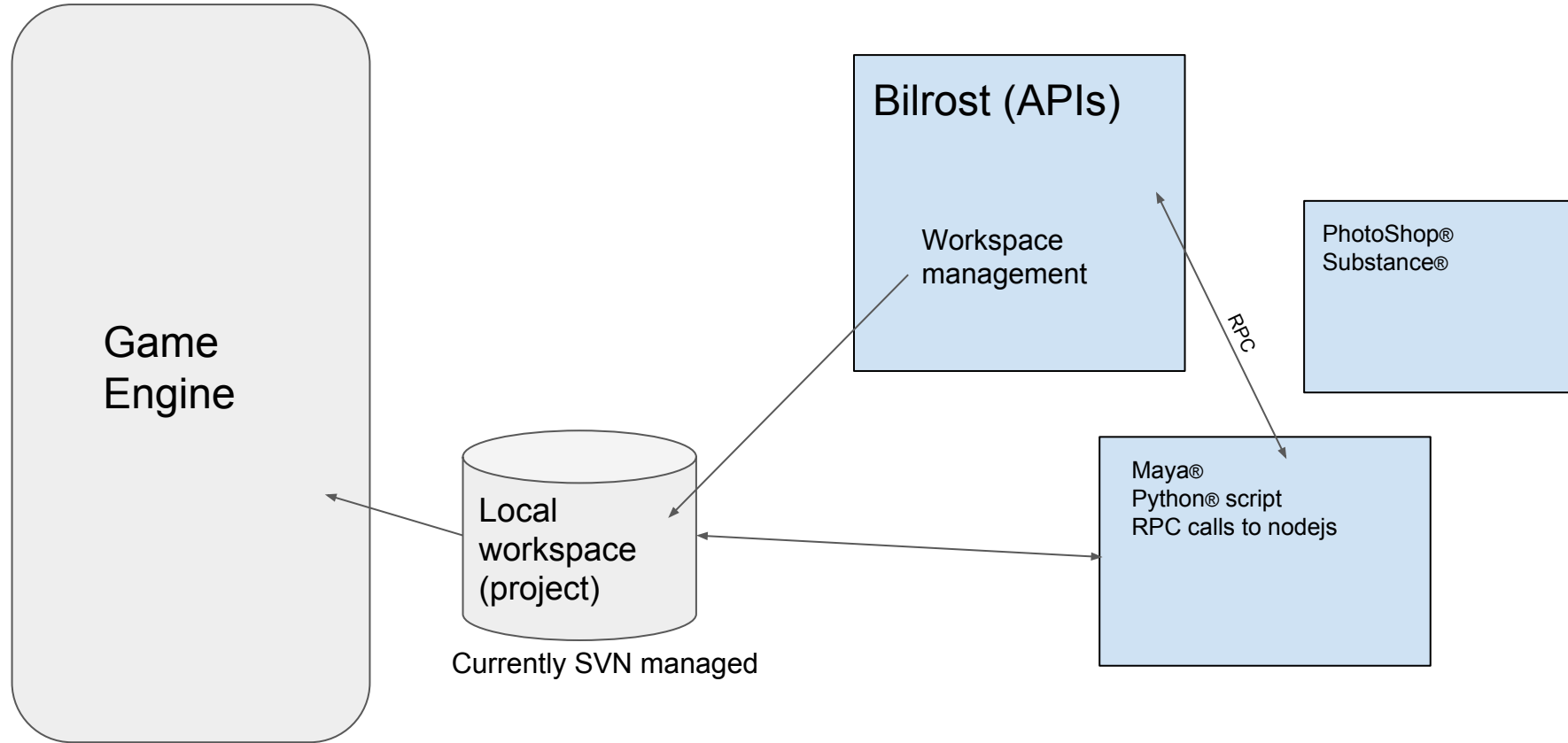
What is Bilrost

- Bilröst is "a rainbow bridge that reaches between Midgard (Human Earth) and Asgard (similar to heaven)." It's not car rust.
- It is a local Node.js server
- It is a framework for different API:
  - Asset Manager/Content Browsing
  - Documentation Wiki
  - Plug-in module for Tools
  - Authentication
  - Others

Bilrost in context

Game Engine

Bilrost (APIs)

Workspace management

PhotoShop®
Substance®

RPC

Local workspace (project)

Currently SVN managed

Maya®
Python® script
RPC calls to nodejs

What is Asset Manager

- It is a set of REST APIs that run on Bilrost
- It is an interface for maintaining and controlling the in-game content and source files used to create the project
- It has a Content Browser

Bilrost Asset Manager API Examples

CB API | LOGIN

API Description

| GET | /assetmanager | Asset Manager's API description + |

WORKSPACES

| POST | /assetmanager/workspaces | Add a Workspace + |
| DELETE | /assetmanager/workspaces/{workspace_guid||workspace_name} | Forget a Workspace + |

STATUSES

| GET | /assetmanager/workspaces/{workspace_guid||workspace_name}/status | Workspace's Statuses + |
| GET | /assetmanager/workspaces/{workspace_guid||workspace_name}/statuses | Workspace's Highest Priority Status + |

ASSETS

PUT	/assetmanager/workspaces/{workspace_guid		workspace_name}{asset_ref}	Create and update an Asset +
POST	/assetmanager/workspaces/{workspace_guid		workspace_name}{asset_ref}/rename	Rename an Asset +
DELETE	/assetmanager/workspaces/{workspace_guid		workspace_name}{asset_ref}	Delete an Asset +

Concepts of Asset and Resource

(What are assets in the context of the Asset Manager)

- Assets are a list of dependencies
- Assets do not contain actual resources
- In fact Assets are a unified way of expressing dependencies

(What are resources in the context of the Asset Manager)

- Resources have two folds:
  * metadata containers with a SHA pointer
  * actual binary content that is stored in the cloud

Asset Manager

What we are trying to solve

- Business models based in frequent DLC releases
- For the long term view refer to Remi's presentations (Asset manager concepts in docs/ folder from open bilrost repo)

What we are trying to solve

Dependencies management

- Start using AM's Assets. That means having a unified representation of dependencies
- why we create an indirection level if dependencies are already in resource files (for example in *.mb files) ?
- Because, given a maya file, it is easy to know its dependencies. But given a .tga it is difficult to know what Maya® files depend on it.
- This problem makes breaking dependencies very easy. Therefore, dependent files are rarely renamed or moved.

What we are trying to solve

Size of repos and location of resources

- Move actual files (resources) from SVN and put them up in the cloud
- Local cache
- Duplicated resources share same SHA1