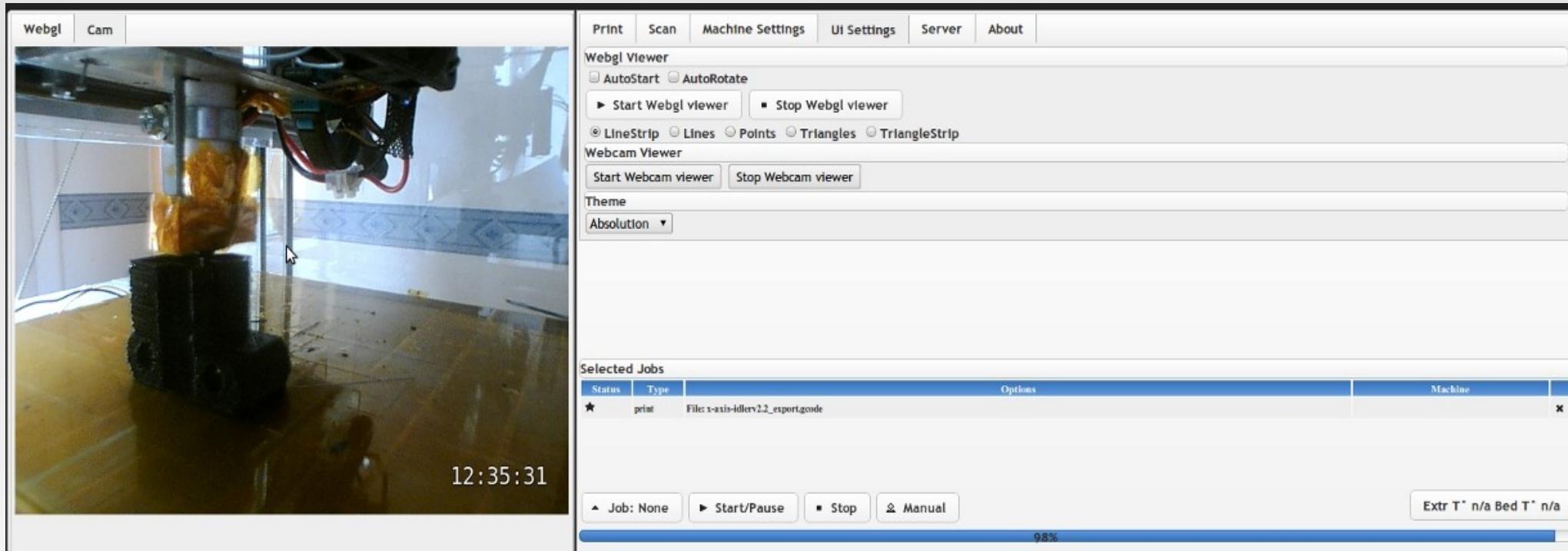


DOBOZ-Web:

Contrôle et monitoring d'imprimantes 3d open source



Mark Moissette kaosat.dev@gmail.com

DOBOZ-Web:

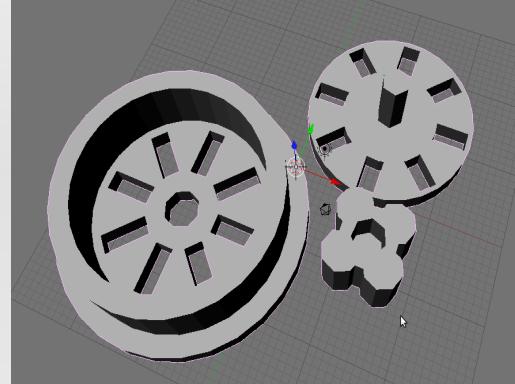
Contrôle et monitoring d'imprimantes 3d open source

- Imprimantes 3d open source
 - Arduino et le projet Reprap
 - Origines du projet Dobozi-Web
- Projet orienté "utilisateur final"
 - Approche générale
 - Processus de développement
 - Implementation initiale "naive"
 - Evolution vers la généricté
 - Post mortem : erreurs , limites et reflexions
- Evolutions futures

Imprimantes 3d open source:

Que fait une imprimante 3d

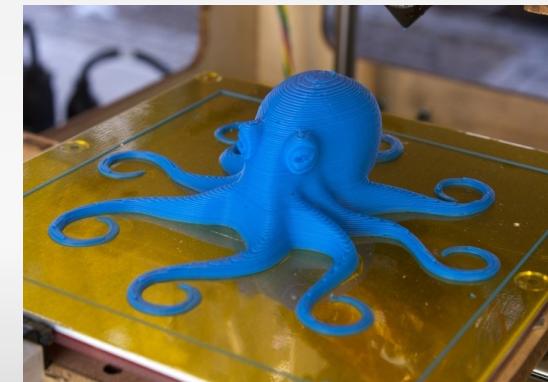
Modelisation 3d



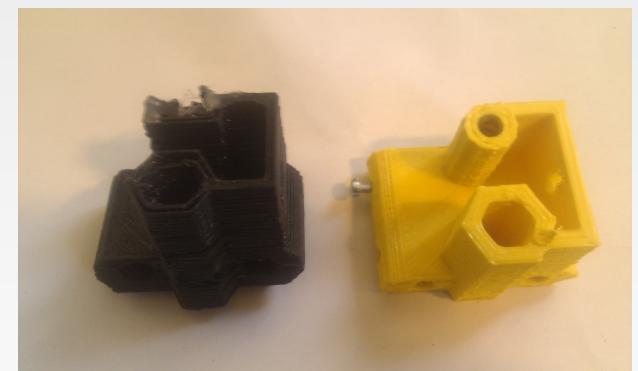
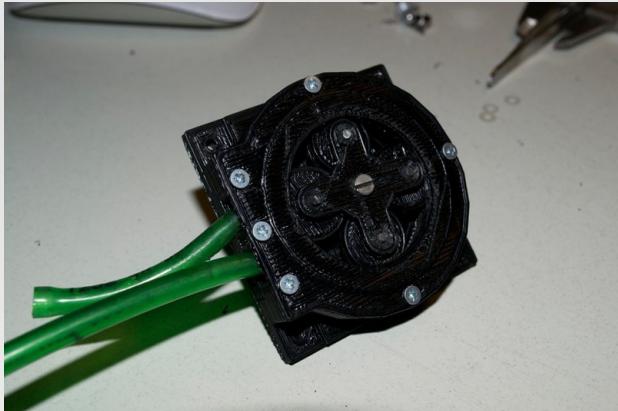
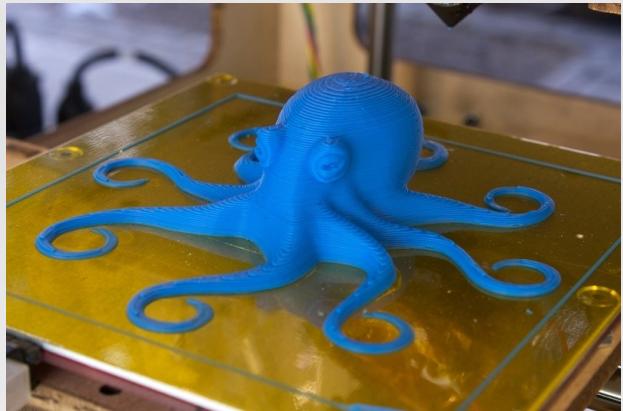
PIPELINE

IMPRIMANTE

Objet physique

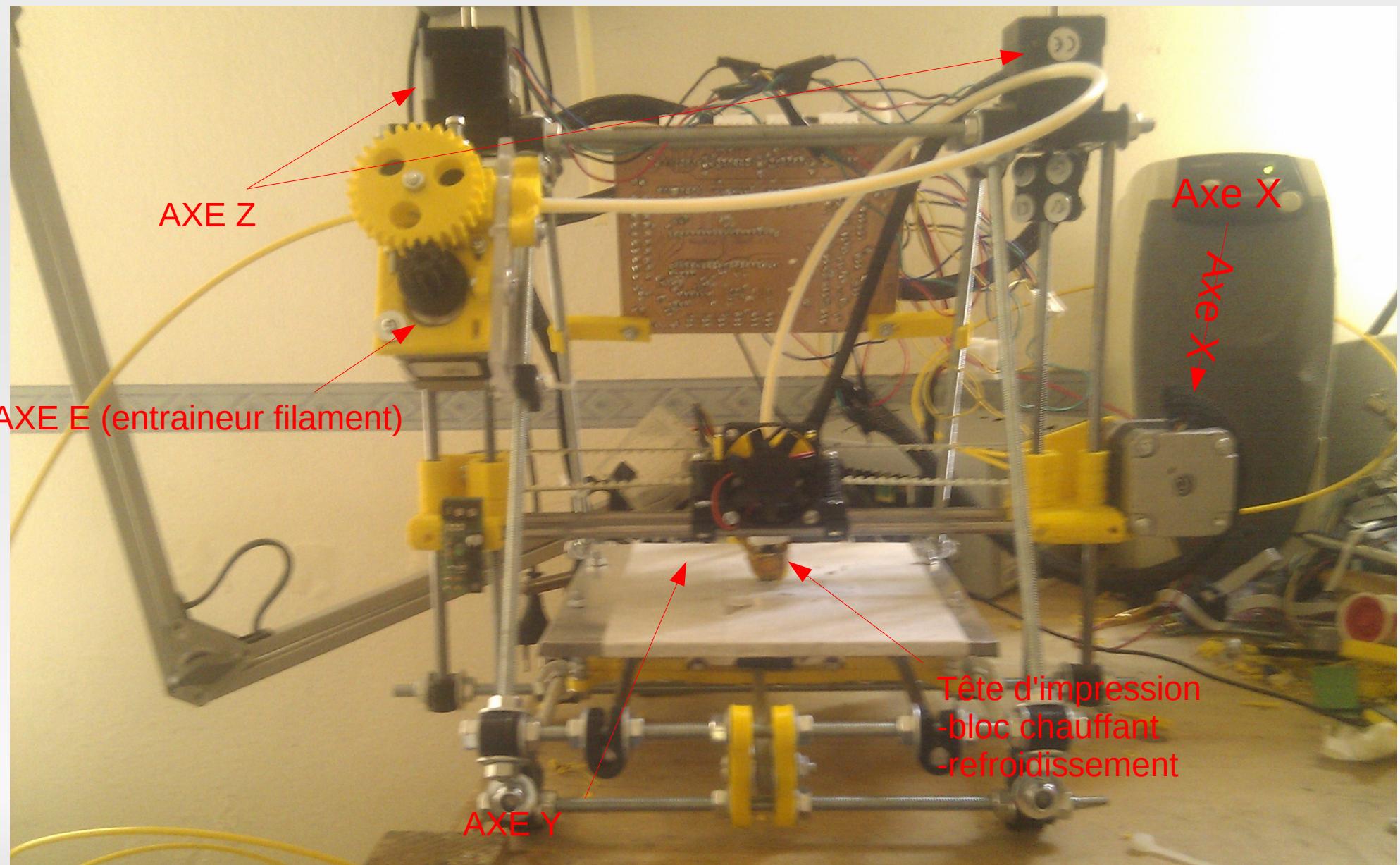


Imprimantes 3d open source: exemples d'objets



Imprimantes 3d open source:

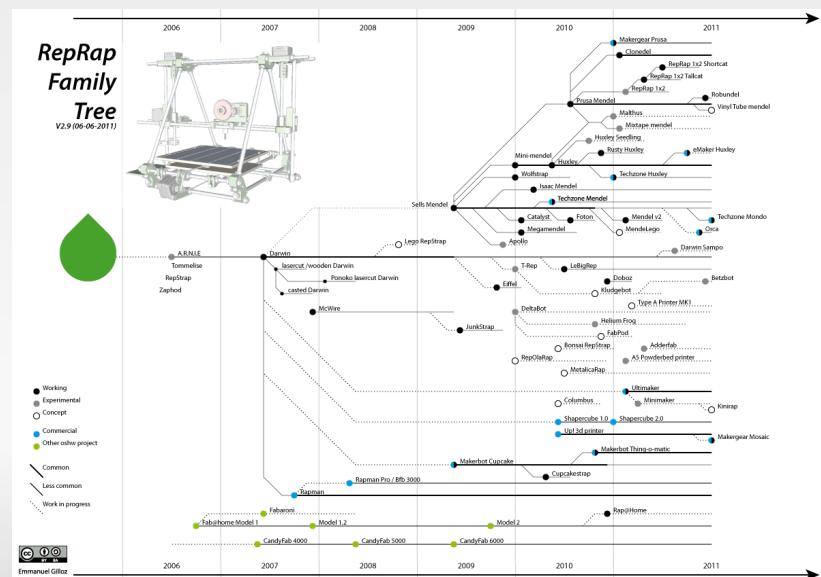
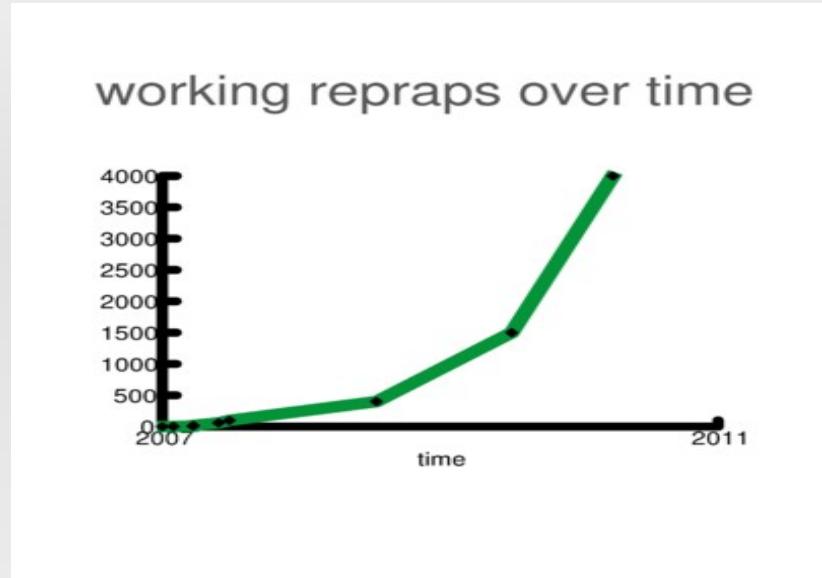
vue d'ensemble d'un Reprap



Imprimantes 3d open source:

Arduino et le projet Reprap: Emergence & explosion

- Arduino:
 - Micro contrôleur open-source
 - Cœur de nombreux projets
 - Axé "grand public"
 - Reprap
 - Imprimante 3d open source
 - "La chine sur votre bureau"
 - 2007 : première machine
 - 2011 :
 - plusieurs milliers de machines
 - Plusieurs dizaines de variantes
 - Possibilités d'évolution quasi infinies



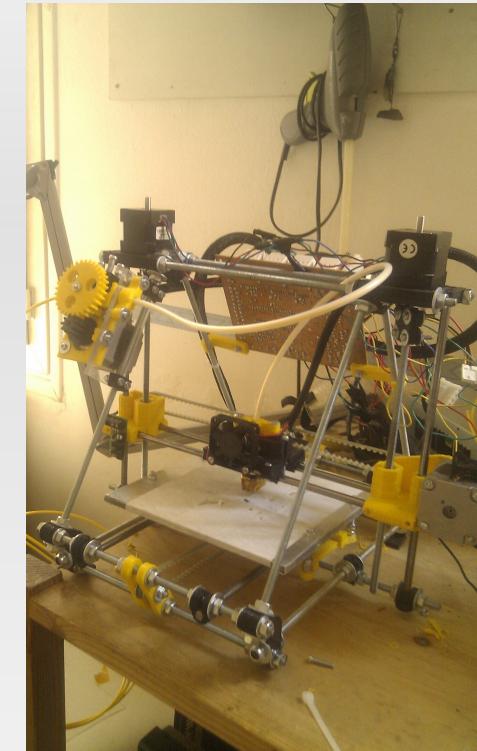
Imprimantes 3d open source:

Arduino et le projet Reprap

- Deux problèmes majeurs:
 - Hétérogénéité :
 - Hardware
 - Software: protocoles etc
 - Offre logicielle peu développée

Outils généraux basiques, ou trop spécialisés

- Skeinforge (Python)
- Replicatorg (Java)



Approche générale:

Des besoins bien définis

- Elements clés :
 - Multi plateforme
 - Accessibilité aux non informaticiens
 - Masquer la complexité
 - Masquer l'hétérogénéité
 - Faciliter le déploiement
 - Installation simple
 - Limitation nombre de bibliothèques
 - Exigences système faible

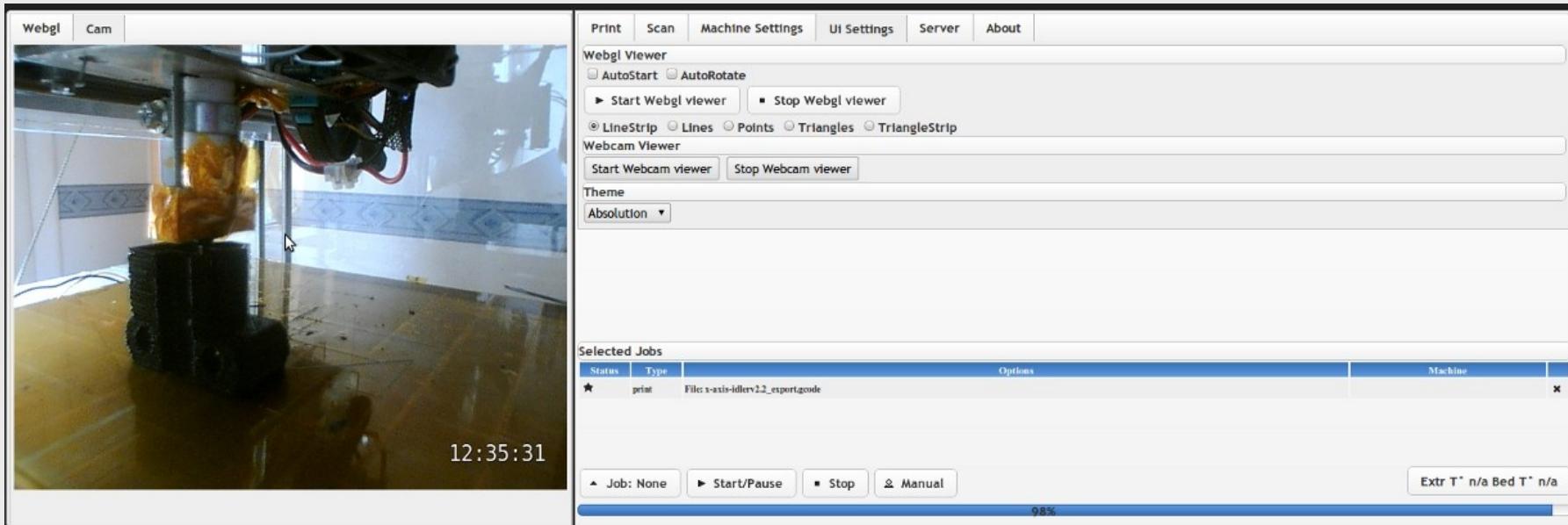
Origines du projet:

Découverte Python & Pyserial

- Envie d'expérimentation interactions host<>contrôleur
- Aspect multi plateformes essentiel:
 - Tentatives sous Mono
 - Premiers pas sous Python
- Pyserial : passage décisif
 - Facilité d'utilisation, fonctionnel , robuste
 - Premier prototype en une journée
 - "Déclic" et adoption de Python!
 - "Hydroduino" (automatisation jardin) premier projet

Approche générale

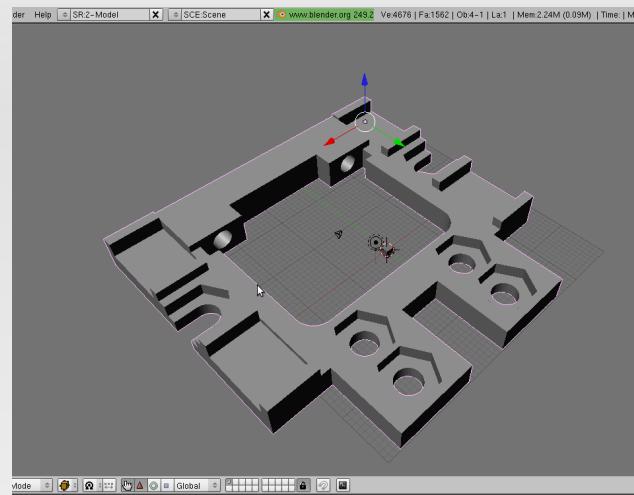
A l'écoute des utilisateurs



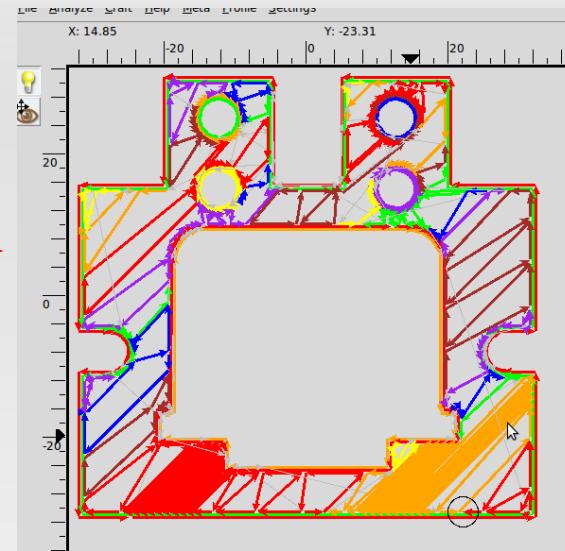
Approche générale

Positionnement du projet dans le pipeline général

Modelisation (Blender etc)



Découpe (Skeinforge)

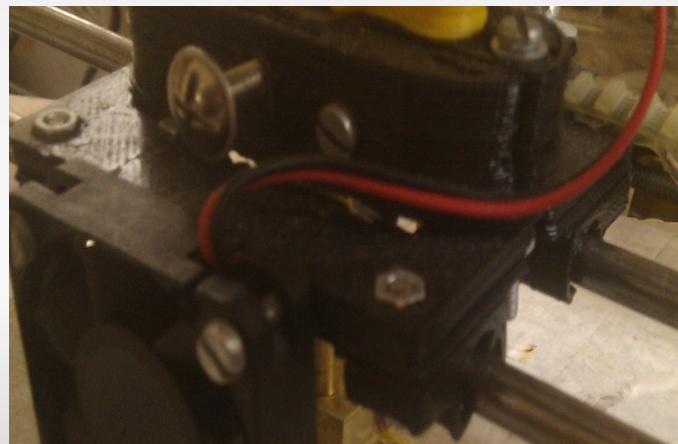


Reglages
Envoi commandes
Monitoring etc



DOBOZ-WEB

Objet physique



Processus de développement:

Vue d'ensemble

- Certains aspects de la "programmation agile"
 - Dogfooding: améliorations et déboggage par utilisation
 - Cycle court de refactorisations
- Approche expérimentale
 - Multiple approches par "feature" puis choix
 - Apprentissage python
- Implementation "naïve" initiale



Généralisation

Processus de développement:

Choix client/serveur

- Raisons multiples
 - Interface utilisateurs multi plateforme :
 - Limitation Gtk/Qt etc sur multi-plateforme
 - Lourdeurs interfaces classiques
 - Web s'impose :
 - multi plateforme
 - Leger
 - Riche ajax, html5, webgl

Processus de développement:

Choix client/serveur (2)

- Possibilités de complexité low cost
 - Sheevaplug
 - Reconversion téléphones portables
- Devenu choix philosophique :
 - Decoupling additionnel
 - Gestion "parc" de machines

Processus de développement:

Richesse des librairies Python: un atout majeur

- Chaque aspect sa librairie:
 - Connection host<>contrôleur: pyserial
 - Web: Bottle : micro framework en un fichier
 - Vidéo : gstreamer /pygst
 - Scons: compilation & upload programmes c++ intégrée

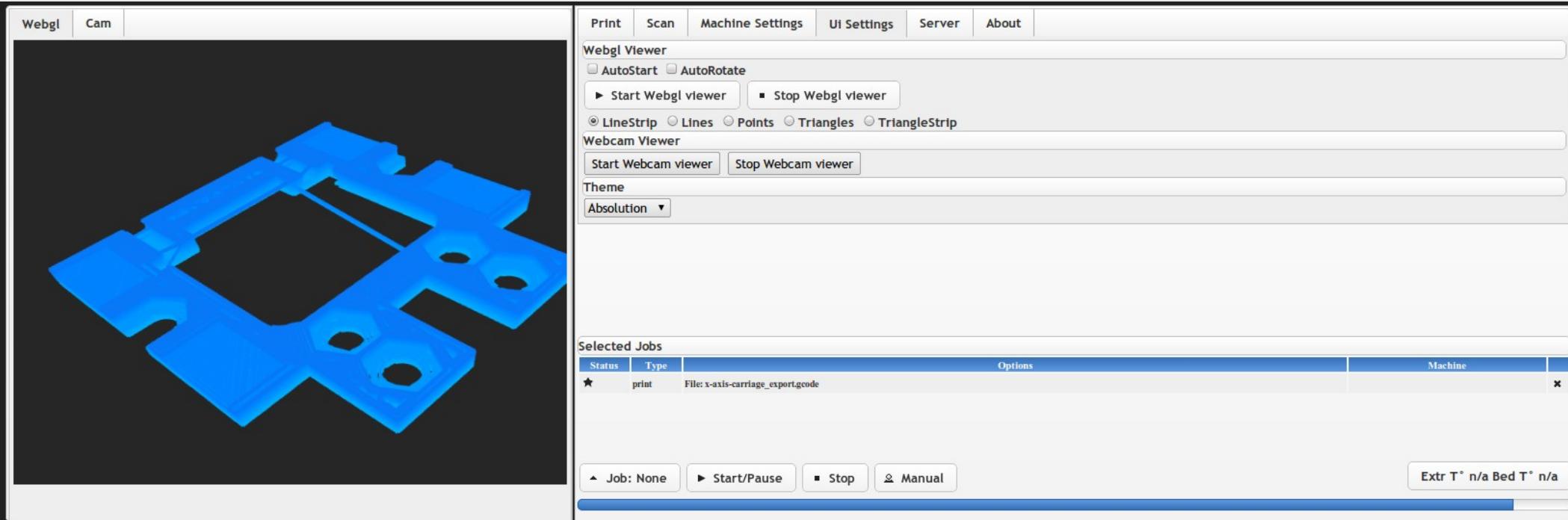
Implementation:

Première version "naïve": objectifs atteints

- Fonctionnalités implementées et fonctionnelles:
 - Contrôle à distance:
 - Parsage et envoi commandes issues de slicer (Gcode)
 - Enchainement d'impressions
 - Scan 3d
 - Commandes manuelles :
 - Déplacement tête d'impression
 - Réglage températures
 - Multi plateformes Win, Linux, Android(client)...
 - Affichage de l'impression (ou scan 3d)
 - 3d temps réel : pseudo streaming + webgl
 - "video" : webcam : qq frames/seconde

Implementation:

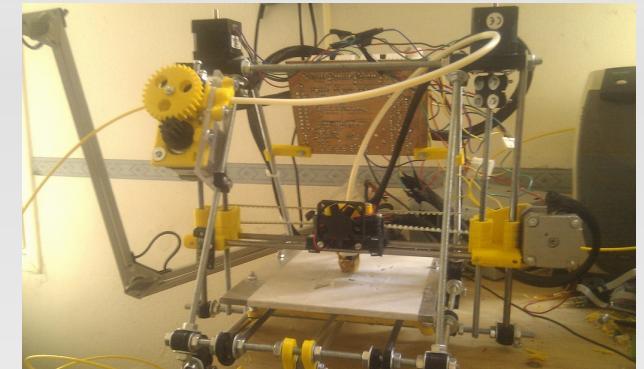
Première version "naive": exemple d'utilisation



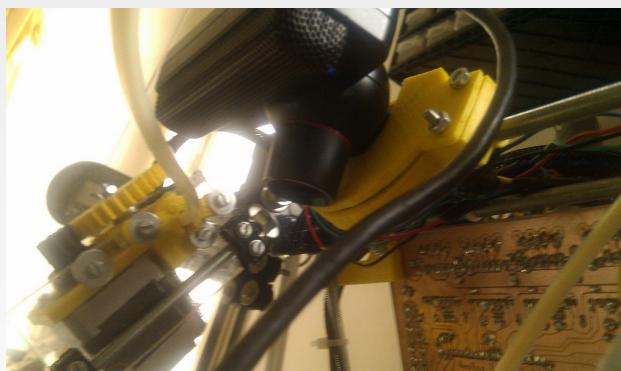
<http://www.youtube.com/watch?v=RvZMali4vAM>

Implementation:

Première version "naïve": résultats



- Utilisation "en production"
- Utile:
 - Utilisable dès les premières version
 - A servi à l'impression d'une machine de seconde génération
- Fiable:
 - A permis de contourner des limitations matérielles



Implementation:

Projet fonctionnel , mais...

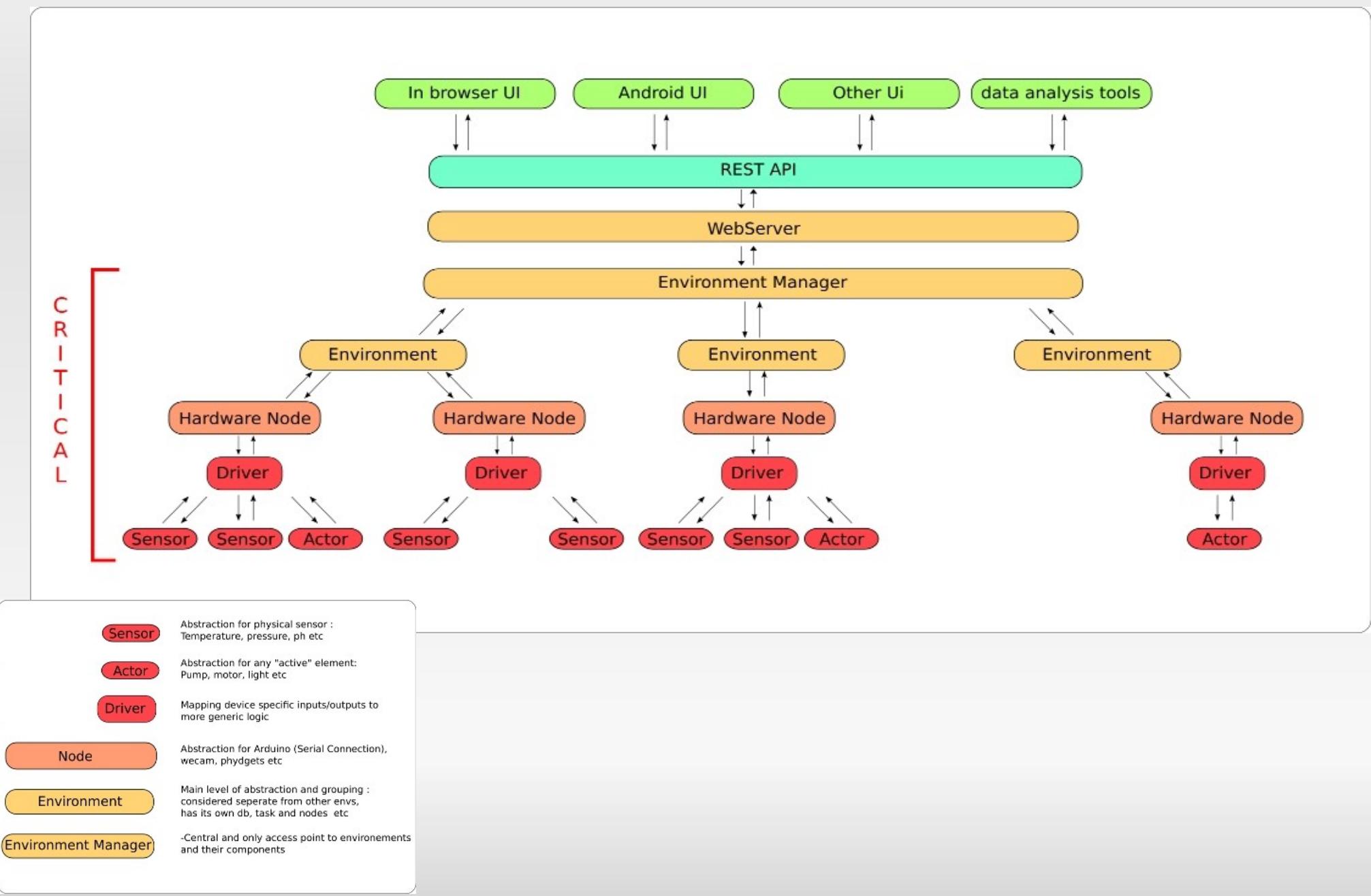
- Pas de généricité:
 - Gestion d'**un seul** type de machine (protocole et firmware fixe)
 - Interactions client serveur type RPC non standardisées
 - Automatisation réduite
 - Aucune optimisation
- Code peu propre
 - Redondances
 - Lourdeurs
 - Multiplication des librairies

Evolution vers la générnicité

- Abstraction des variations matérielles et logicielles
- Modélisation logicielle d'objets physiques pour faciliter leur manipulation
- Acteurs : manipulent des grandeurs physiques (position, température, pression etc)
- Capteurs : retour d'information sur ces même grandeurs physiques
- Noeud hardware : abstraction d'une machine
Regroupe capteurs & acteurs

Evolution vers la générnicité

Vue d'ensemble



Post mortem:

Revers de l'approche orientée utilisateur

- Complexité déportée vers le développement
- Recherche du "saint graal" d'unification de machines (reprap) hétérogènes
 - Pas forcément possible ?
 - Où gérer le passage spécifique → générique ?
 - Blocages conceptuels multiples
- Machines et protocoles en évolution constante!
 - Manque de vue d'ensemble
 - Manque de standardisation

Post mortem:

le déploiement

- Déploiement d' **applications** en Python:
 - Plus axé développeurs qu'utilisateurs?
- Plusieurs solutions envisagés:
 - Builds par plateforme
 - Standalone (freeze etc)
 - ZCBuildout (+VirtualEnv) ←
- Reste problématique

Post Mortem:

complexité excessive

- Projet trop ambitieux
 - Complexité accrue par exigences de base
 - "Feature creep" : manque de définition claire des fonctionnalités
 - Refactorisations à outrance
- Réinventer la roue:
 - ok pour apprentissage
 - non adapté à project complexe

Evolution future

- Faciliter l'évolution du logiciel
 - Modularité
 - Généricité
- Faciliter son adoption
 - Documentation, ouverture
 - Développement orienté tests
- Migration vers Python Twisted (asynchrone, haute performance)
 - Gestion multi protocoles adaptée
 - Limiter le nombre de librairies
 - Arreter de réinventer la roue

Evolution future: Un pas de plus vers la générnicité

- Problématique et objectif ultime de générnicité:
 - Gestion d'un ensemble de machines hétérogènes par un système unifié
 - Integration d'un ensemble d'applications (workflow variables)
- Extraire les éléments comuns
 - Coeur
 - Plugins/addons
- Trouver le bon "paradigme"
 - Faciliter l'ajout de machines
 - Encapsuler les différences

Evolution future:

Approche descriptive

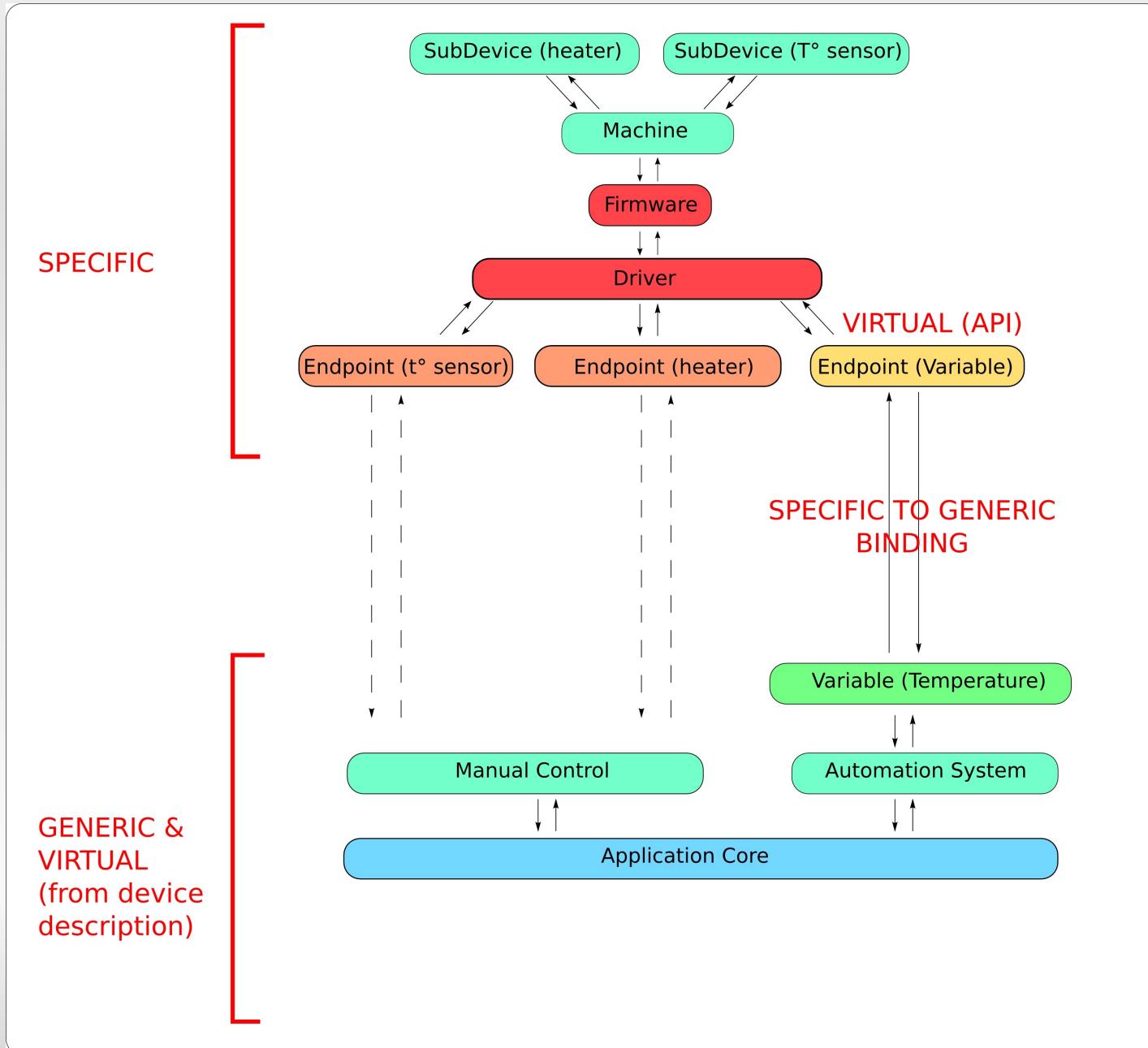
- Description de machine en format texte (json)

```
"PrintHead1": {  
    "parent": "PositioningTool",  
    "variables": [  
        {  
            "name": "head_temp",  
            "type": "température",  
            "units": "celcius"  
        },  
        {  
            "name": "filament extrudate",  
            "type": "position",  
            "components": "e1",  
            "units": "mm"  
        }  
    ],  
    "elements": [  
        {  
            "name": "e stepper",  
            "type": "stepper_motor",  
            "var": "filament extrudate",  
            "var_channel": "e"  
        },  
        {  
            "name": "extruder1_heater",  
            "type": "heater",  
            "var": "head_temp"  
        },  
        {  
            "name": "extruder1_fan",  
            "type": "cooler",  
            "var": "head_temp"  
        },  
        {  
            "name": "extruder1 temp sensor",  
            "type": "temperature",  
            "components": "e1",  
            "units": "celcius"  
        }  
    ]  
}
```

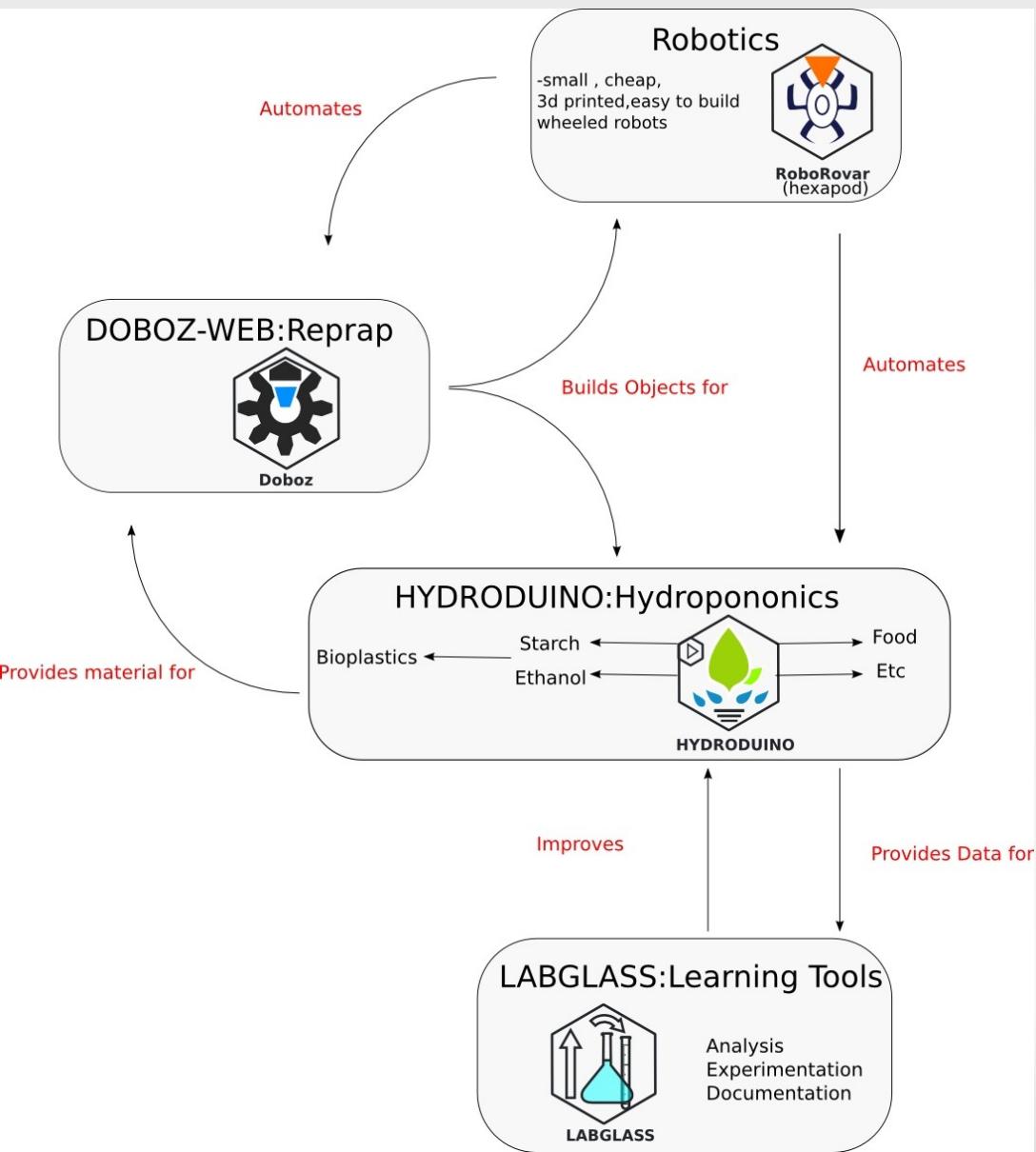
```
{  
    "node": {  
        "name": "my_reprap",  
        "description": "a_clara_reprap",  
        "tools": [  
            {  
                "PositioningTool": {  
                    "variables": [  
                        {  
                            "name": "3d_position",  
                            "type": "position",  
                            "components": "x,y,z,f",  
                            "units": "mm",  
                            "master_variable": true  
                        }  
                    ],  
                    "elements": [  
                        {  
                            "name": "x_stepper",  
                            "type": "stepper_motor",  
                            "var": "3d_position",  
                            "var_channel": "x"  
                        },  
                        {  
                            "name": "y_stepper",  
                            "type": "stepper_motor",  
                            "var": "3d_position",  
                            "var_channel": "y"  
                        }  
                    ]  
                }  
            }  
        ]  
    }  
}
```

Evolution future:

Du particulier au général



Evolution future: intégration d'applications multiples



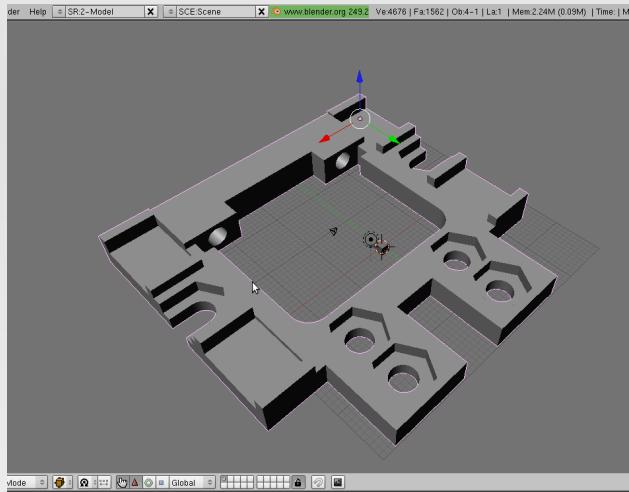
■ A terme:

- **Pollapli**: plateforme de monitoring & contrôle pour machines basées sur arduino
- **DOBOZ-Web** -> plugin

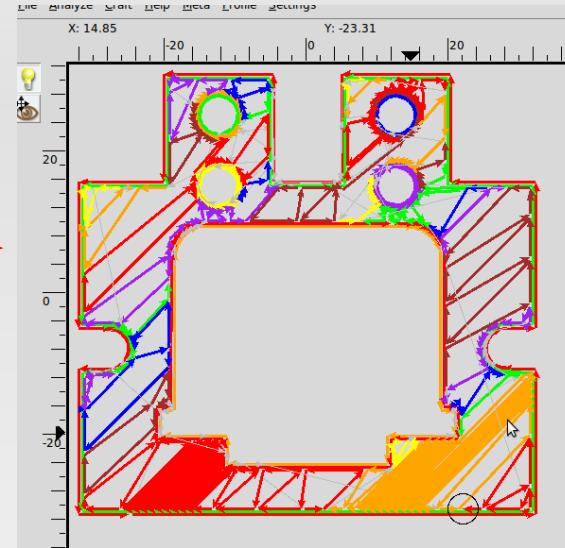
Approche générale

Intégration dans workflow

Modelisation (Blender etc)



Découpe (Skeinforge)

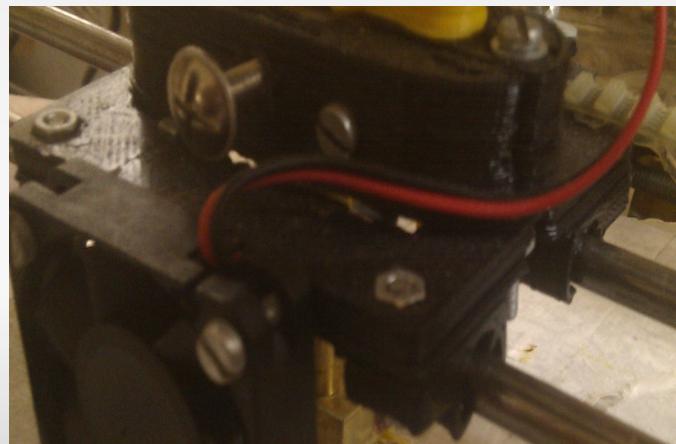


Réglages
Envoi commandes
Monitoring etc



DOBOZ-WEB

Objet physique



Liens et Références

- Merci de votre attention, feedbacks, suggestions et commentaires sont les bienvenus!
- Code téléchargeable et infos:
 - <https://github.com/kaosat-dev/Doboz>
 - <http://reprap.org/wiki/Doboz-Web>
 - <http://www.kaosat.net/?tag=dobozweb>
- Sites officiels Arduino et Reprap:
 - <http://www.arduino.cc/>
 - http://reprap.org/wiki/Main_Page
- Librairie open source d'objets 3d imprimables:
 - <http://www.thingiverse.com/>