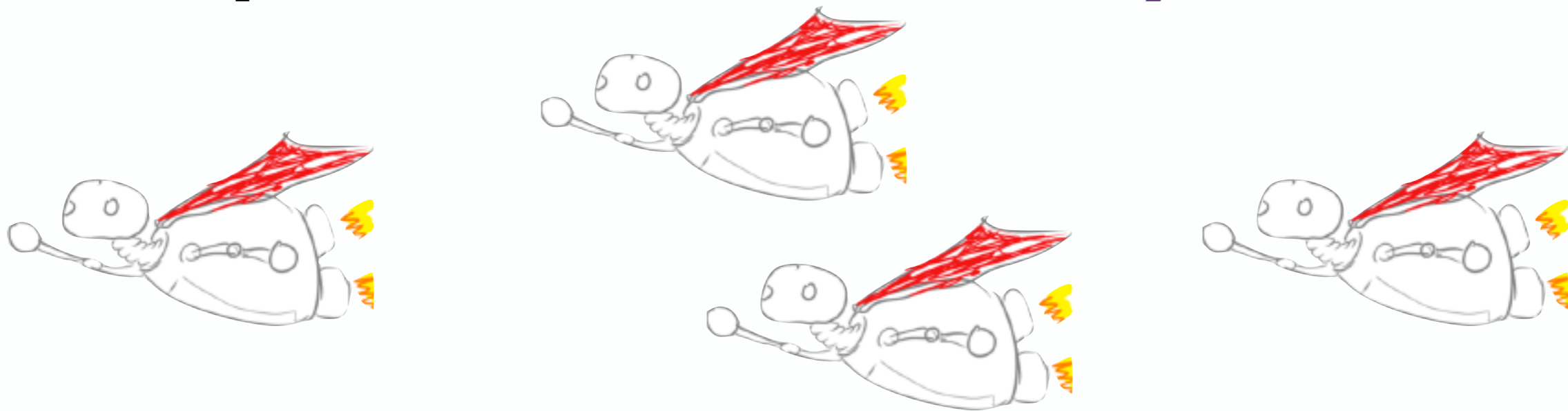


Accelerating SE research adoption with Analysis Bots



<https://github.com/AnalysisBotsPlatform>

Ivan Beschastnikh, Mircea F. Lungu, Yanyan Zhuang



U. of British
Columbia
Canada



U. of Groningen



U. of Colorado
Colorado Springs

Wanted: SE research impact

- SE research community developing numerous analysis techniques and tools that use them
- But, practitioner adoption is a struggle

“many tools are so hard to configure, they prevent you from doing anything.” [1]

“I find that the information they provide is not very useful, so I tend to ignore them.” [1]

“if it disrupts your flow, you’re not gonna use it.” [1]

[1] Brittany Johnson et al. “Why Don’t Software Developers Use Static Analysis Tools to Find Bugs?” ICSE 2013

Each paper creates its own zoo

- (Stereotypical) SE research evaluation workflow:
 - Identify a set of projects/revisions
 - Run the tool on revisions
 - (Optional) Submit bug reports to projects e.g., [1]
 - Write evaluation section



[1] Boyuan et al. "Characterizing and Detecting Anti-patterns in the Logging Code" ICSE 2017

Each paper creates its ~~own~~ petting zoo



 **Bitbucket**

 **GitHub**

Each paper creates its ~~own~~ petting zoo



 **Bitbucket**

- Tool runs on project in controlled environment
- No involvement of the project community
- Tool not used by target user population



GitHub

Each paper creates its own ^{petting} zoo



End-result: slow tool adoption
and low research impact



- No involvement of the project community
- Tool not used by target user population

Going into the wild



Going into the wild: it's hard!

- How will developers find the research tool?
 - Requires evangelism/advertising
- How will developers use the research tool?
 - Requires end-to-end working artifact
- How will they **want** to use the tools?
 - Easy to install
 - Simple to configure
 - Immediate value
 - ...



Going into the wild: it's hard!

- How will developers find the research tool?
 - Requires evangelism/advertising
- How will developers use the research tool?
 - Requires end-to-end working artifact
- How will they **want** to use the tools?
 - Easy to install
 - Simple to configure
 - Immediate value
 - ...



Rethinking SE research transfer

- How do we make it easier (for researchers)?

Rethinking SE research transfer

- How do we make it easier (for researchers)?
...What if developers came to us?



Rethinking SE research transfer

- How do we make it easier (for researchers)?
...What if developers came to us?

Idea: Build a one-stop-shop **platform** for hosting research tools (analysis bots)

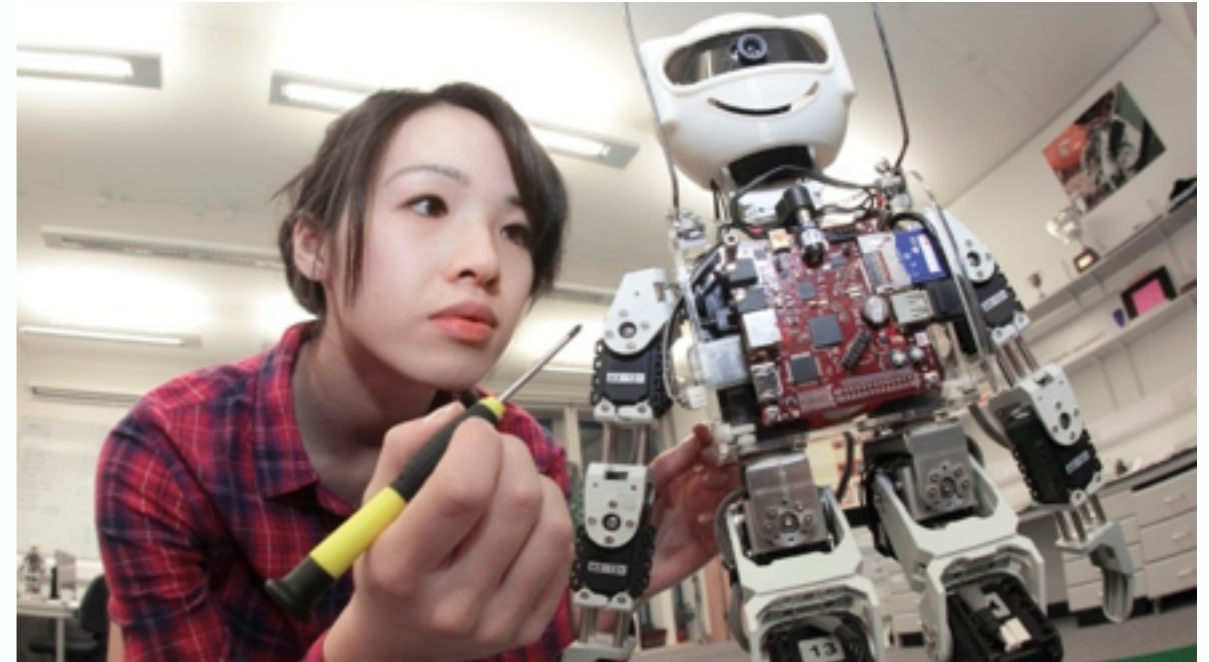


Analysis bots platform vision

Analysis bot := partial automation of some SE task

Key features

- Low barrier to entry
 - For bot users/developers
- Fine-grained bot control
 - Developer in control
- Openness
 - Anyone can use/build a bot



Analysis bots platform vision

Analysis bot := partial automation of some SE task

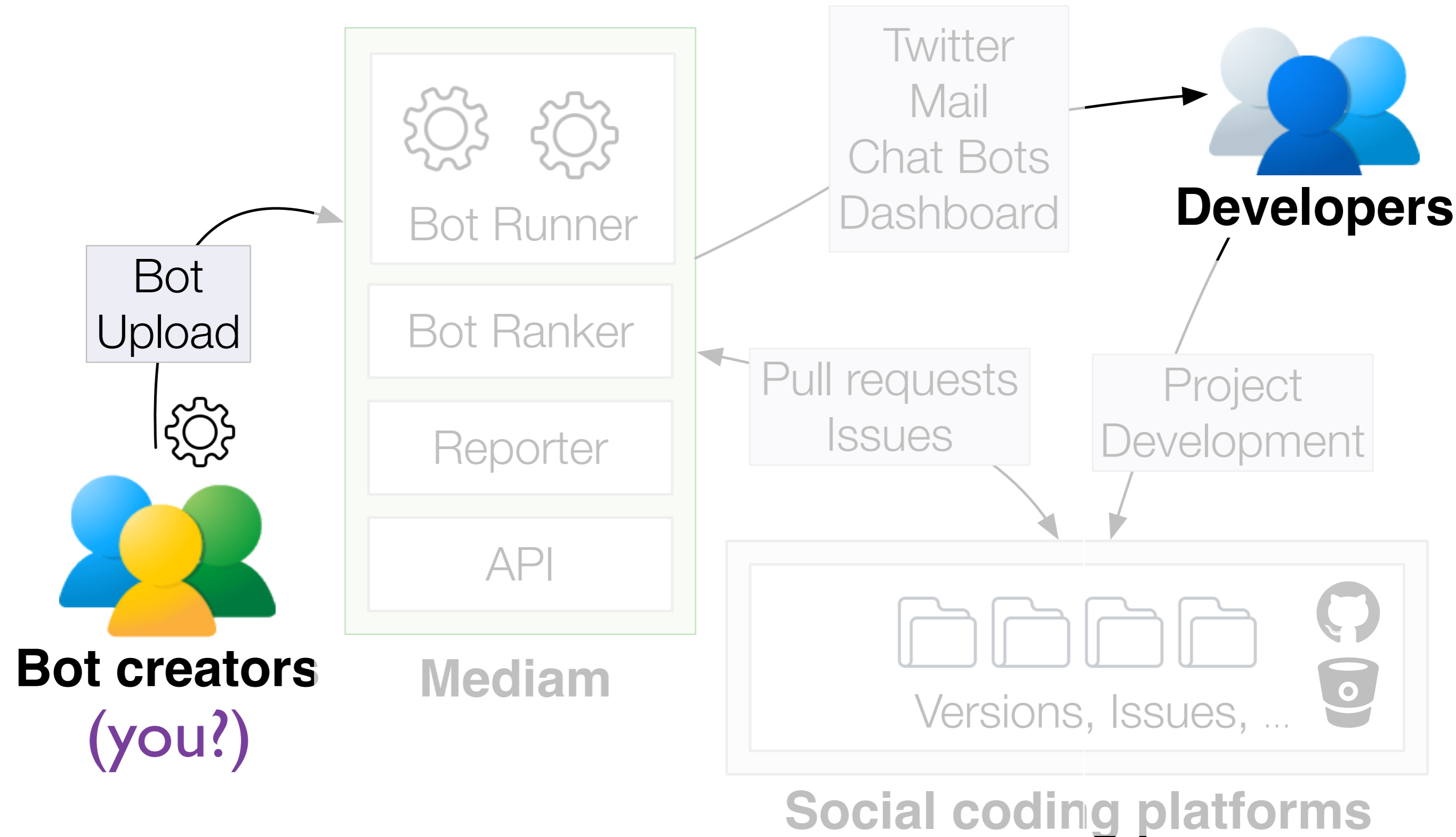
Key features

- Low barrier to entry
 - For bot users/developers
- Fine-grained bot control
 - Developer in control
- Openness
 - Anyone can use/build a bot

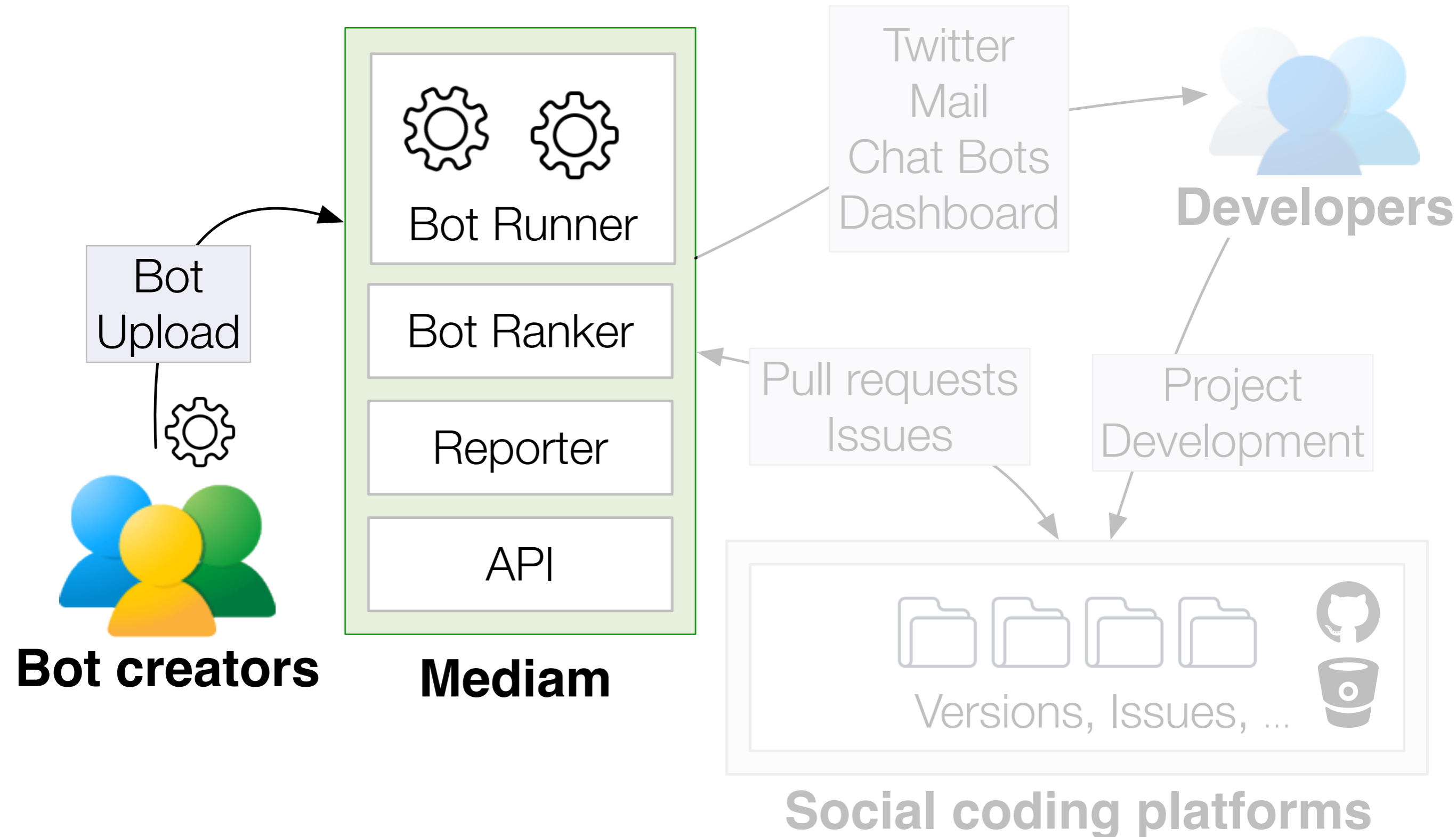
...like iRobot!



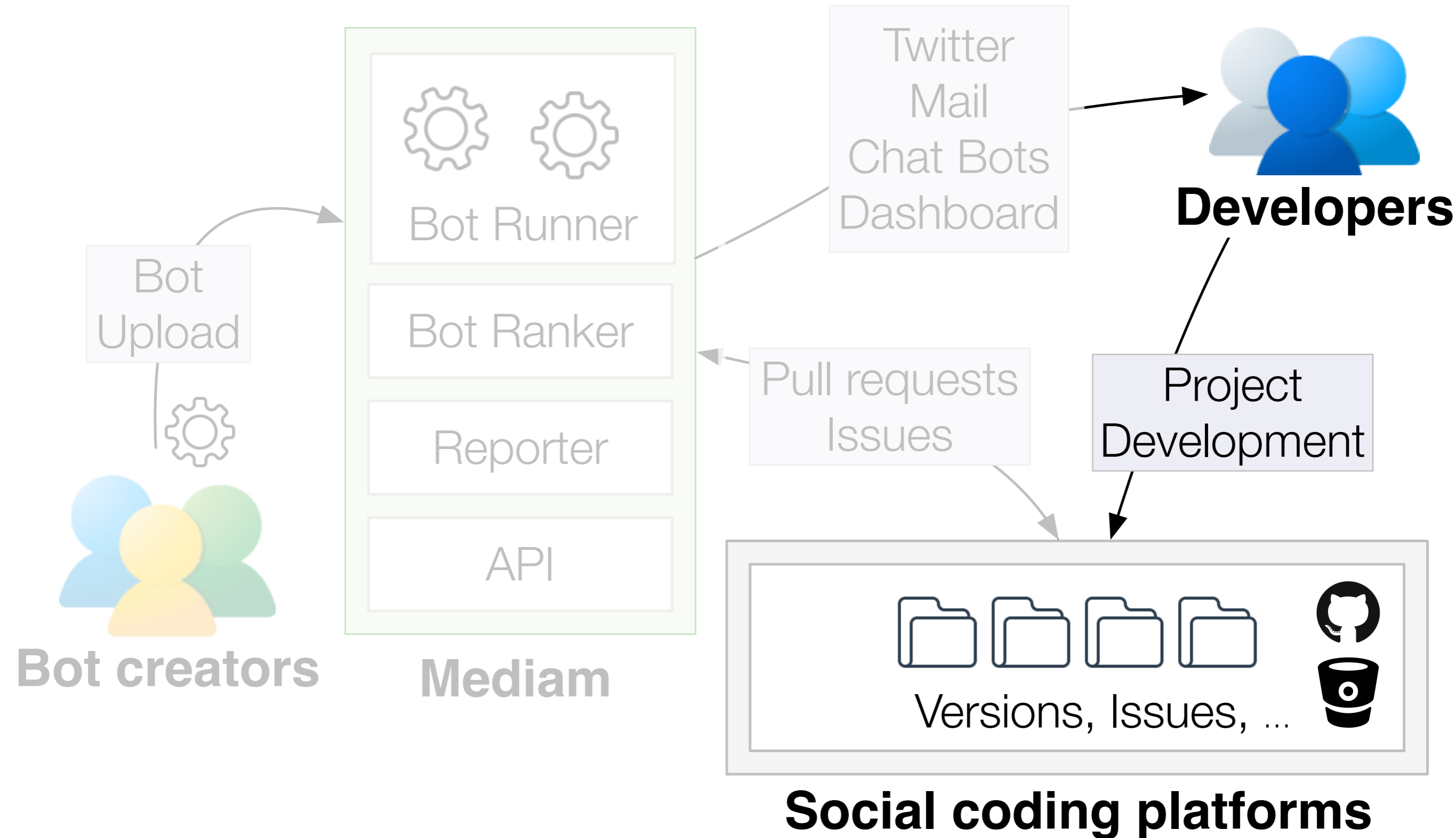
How would it work?



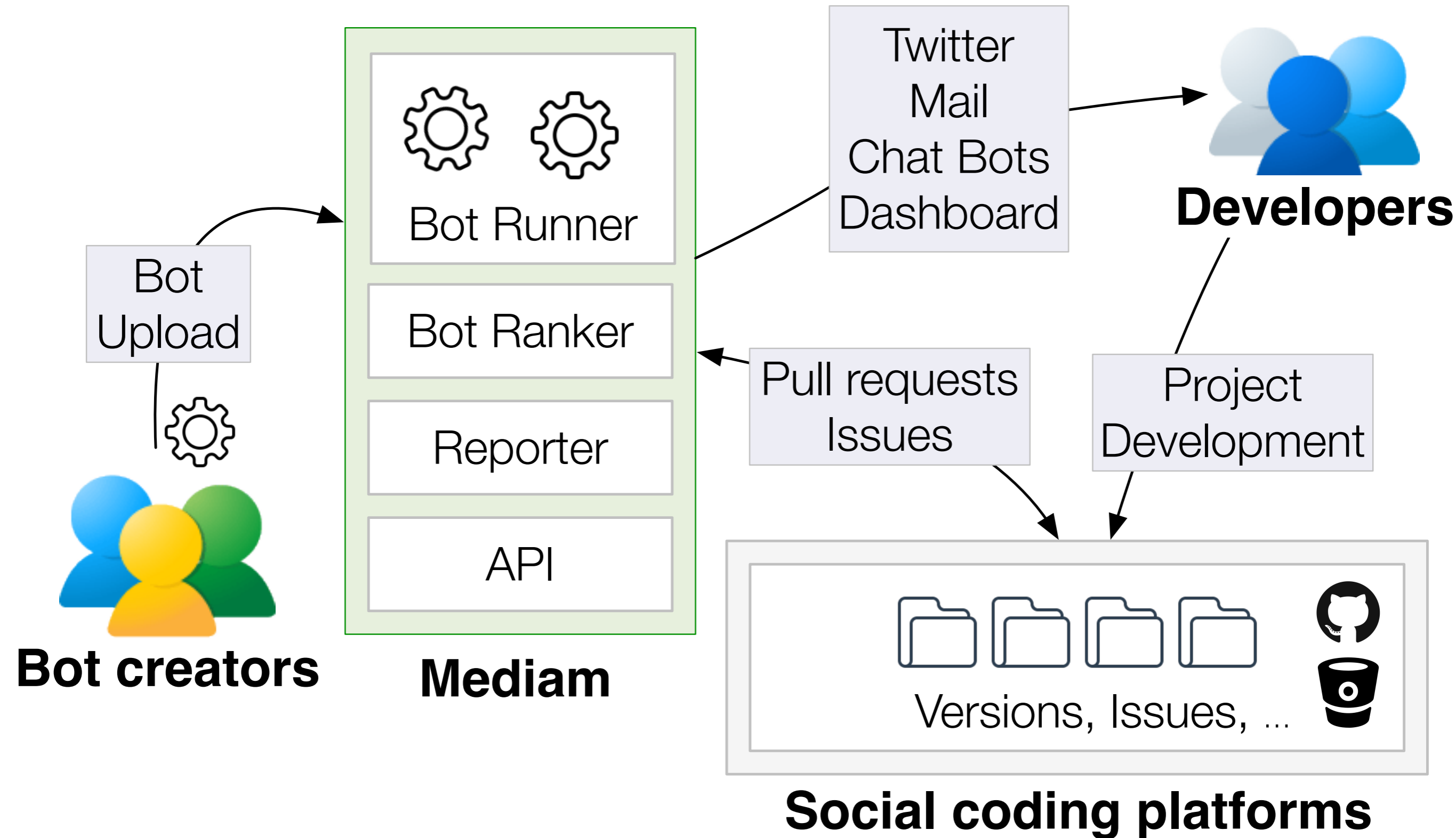
How would it work?



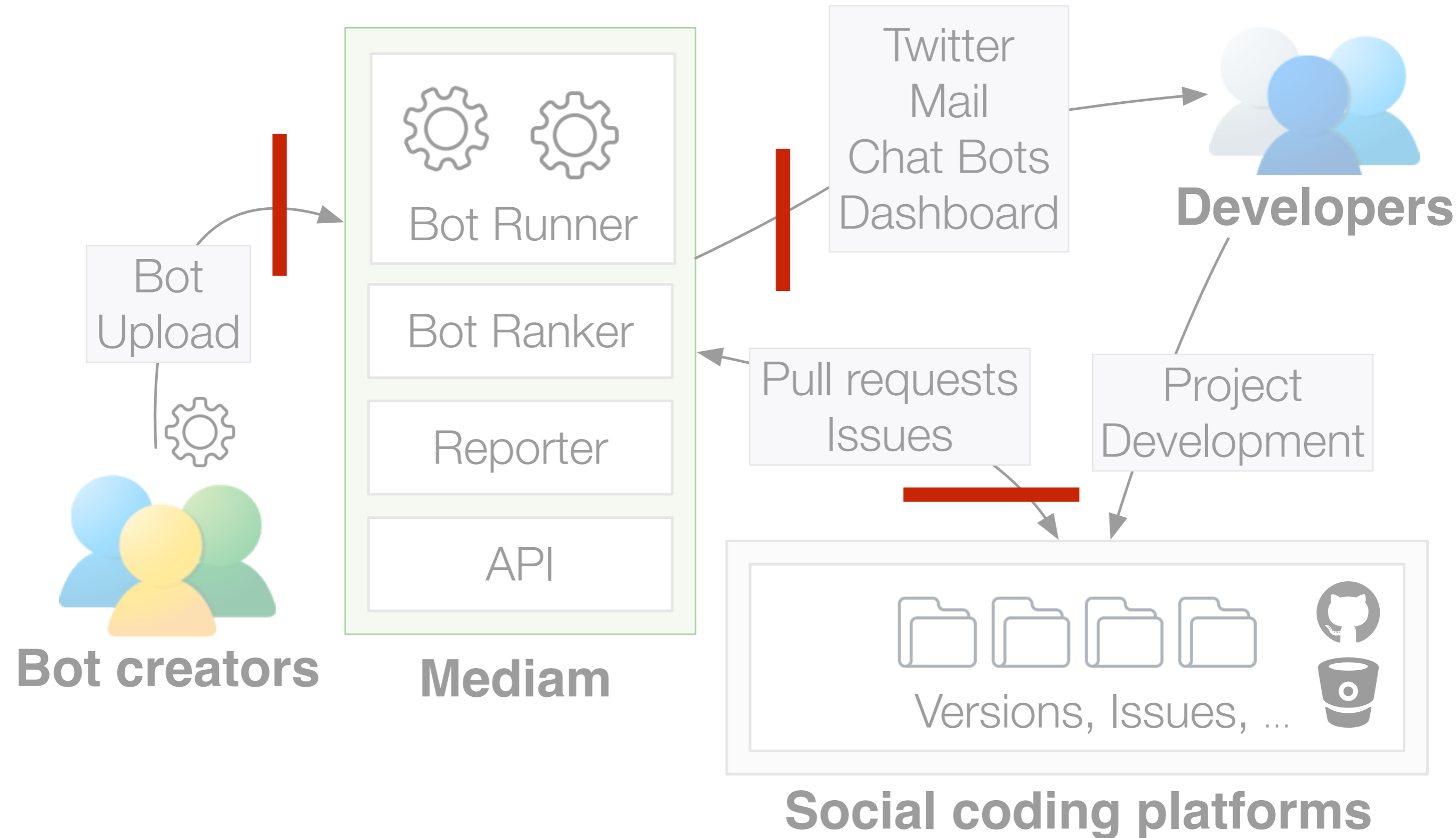
How would it work?



How would it work?



Defining the right APIs is key



Analysis bots benefits

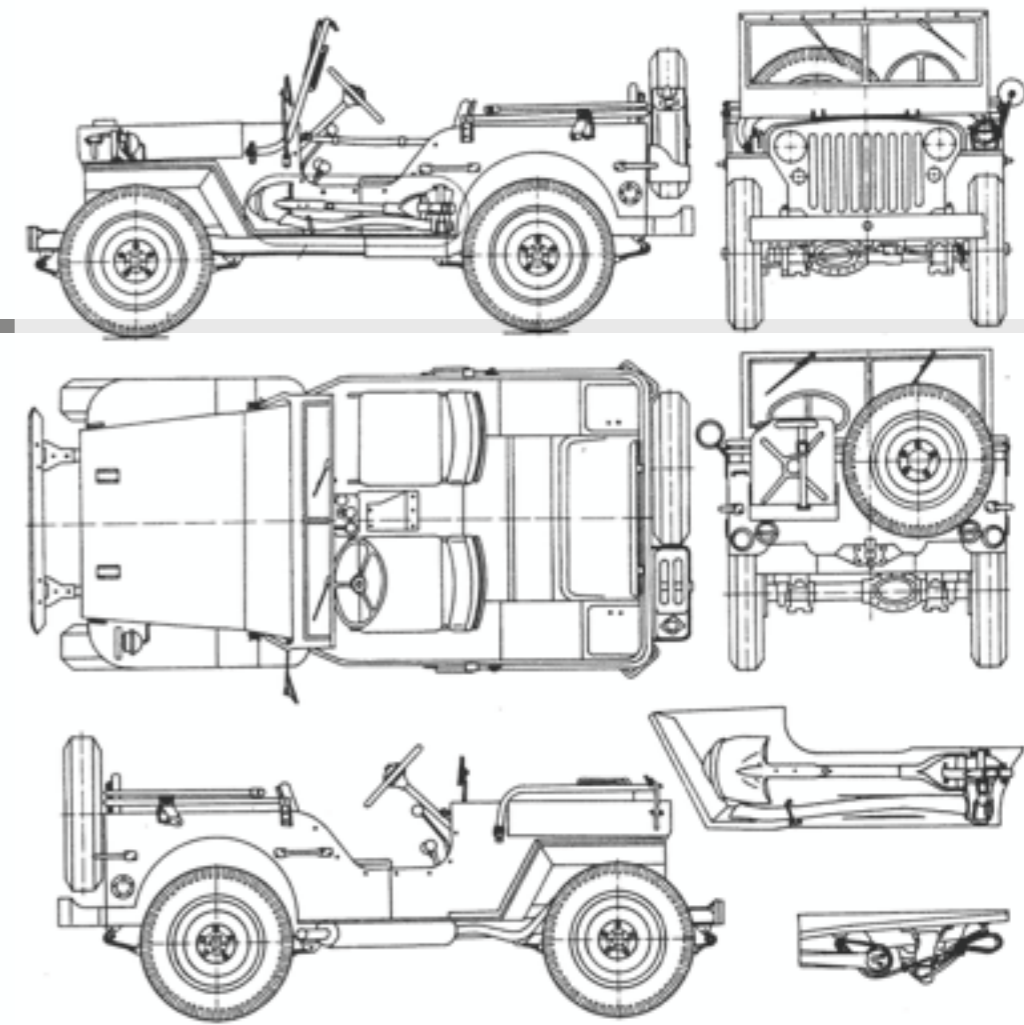
- *Naturalistic* evaluation
- Improved replication of academic artifacts
- Deals with aspects of tool delivery that tool developers don't want to deal with
- Quick iteration to understand what works and what doesn't
- Developers get to subscribe and use the latest and greatest SE research

(Survival of the fittest: most useful tools receive attention and contributions)

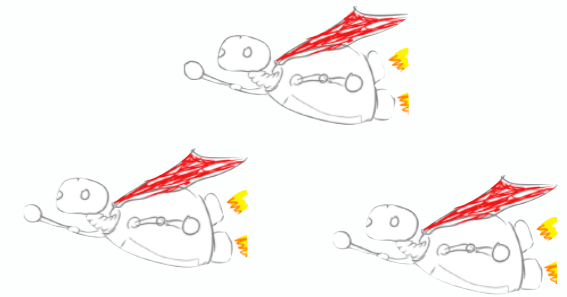


Open questions

- **Trust/discovery**: how to bootstrap uptake of the platform?
- **Preventing abuse**: what are the appropriate and inappropriate ways in which analysis bots can interact with developers?
- **Usability**: how can analysis bots be designed to prevent information overload for developers?
- **Economics**: How and where should these bots be hosted?
- **Design**: What are the right platform APIs and reputation system?



Related work



- **Google:** Tricorder [1] and Shipshape [2] (analysis infrastructure)
- **Academia:** CodeDrones [3]
- **More academia:** Testing-as-a-service [4] and AppInspector [5]
- **OpenSource:** Various bots on GitHub; Imageoptimiser bot [6]
- **Other ‘bots’:** Hubot programmable bot [7]

[1] Sadowski et al. Tricorder: Building a Program Analysis Ecosystem. ICSE 2015

[2] <https://github.com/google/shipshape>

[3] Acharya et al. Code Drones. ICSE 2016

[4] Candea et al. Automated Software Testing As a Service. In SoCC, 2010.

[5] Gilbert et al. Automated Security Validation of Mobile Apps at App Markets. MCS, 2011

[6] <https://github.com/imageoptimiser>

[7] <https://hubot.github.com/>

Analysis bots vision



SE research adoption a consistent challenge; requires concentrated effort

Vision: Analysis Bots platform lowers barriers for developers and researchers

- Bot discovery
- Trust
- Information overload
- + Centralized repository
- + A robust reputation system
- + Common APIs and runtimes

<https://github.com/AnalysisBotsPlatform>