

# UC Berkeley

## Anthropology Faculty Publications

### Title

AppCivist - A Service-Oriented Software Platform for Socially Sustainable Activism

### Permalink

<https://escholarship.org/uc/item/2013j8m0>

### Journal

Proceedings - International Conference on Software Engineering, 2

### ISSN

0270-5257

### ISBN

9781479919345

### Authors

Pathak, A  
Issarny, V  
Holston, J

### Publication Date

2015-08-12

### DOI

10.1109/ICSE.2015.185

Peer reviewed

# AppCivist - A Service-oriented Software Platform for Socially Sustainable Activism

Animesh Pathak\*, Valérie Issarny\*, and James Holston†

\*Inria, France — Email: firstname.lastname@inria.fr

†University of California, Berkeley, CA, USA — Email: jholston@berkeley.edu

**Abstract**—The increased adoption of mobile devices and social networking is drastically changing the way people monitor and share knowledge about their environment. Here, information and communication technologies (ICT) offer significant new ways to support social activism in cities by providing residents with new digital tools to articulate projects and mobilize activities. However, the development of ICT for activism is still in its infancy, with activists using basic tools stitched together in an ad hoc manner for their needs. Still, Internet-based technologies and related software architectures feature various enablers for civic action beyond base social networking. To that end, this paper discusses the vision and initial details of AppCivist, a platform that builds on cross-domain research among social scientists and computer scientists to revisit service-oriented architecture and relevant services to further social activism. We discuss the ICT challenges inherent in this project and present our recent work to address them.

## I. INTRODUCTION

The latest Information and Communication Technologies (ICT) enable citizens to get actively involved in shaping and monitoring their lived environment in unprecedented ways. By providing residents with new digital tools to articulate projects, deliberate directly among themselves, and mobilize activities, ICT may support significant social activism and promote social change [1]. Leveraging ICT for social activism is, however, still at an early stage of development in which activists use base rather than specialized tools—most notably social networks and Web-based content management systems—to address their needs. As a result, the challenges of ICT development for social activism is not as much to provide the support building blocks as to help activists pinpoint the relevant Internet components and to compose them in a way that meets their specific requirements. As these demand new kinds of Internet configurations significantly different from traditional Internet-based systems, the challenges are considerable. Enabling the composition of Internet-scale applications for social activism is specifically the goal of the software platform AppCivist,<sup>1</sup> a project of collaborative research between anthropologists and computer scientists.

AppCivist builds upon the experience of the Social Apps Lab<sup>2</sup> that focuses on interdisciplinary collaboration to create mobile and web applications that encourage citizen participation, urban knowledge, and crowdsourcing solutions for

significant social problems. Other applications developed by the Social Apps Lab include Dengue Torpedo<sup>3</sup> which promotes citizen engagement to report and eliminate mosquito breeding sites. This background provides us with insights about key functionalities that need to be accounted for in applications for social activism. Based on this knowledge, the design of AppCivist aims to develop, from prototype implementation to city-scale experiment, a software platform that lets activist users compose their own applications, called *Assemblies*, using relevant Internet-based components enabling democratic assembly and collective action.

Starting from the social science perspective [2], we identify the following categories of functions for *Assemblies*:

*Involving, reaching and mobilizing*: The first step of social activism lies in reaching a relevant community of activists and prompting their involvement. While modern communication means like social networking allow communicating at a global scale, they also tend to overwhelm and distract people. However, traditional social activism relies on a possibly small but largely dedicated group of people. The challenge for an *Assembly* is thus about reaching the relevant population of users through incentives that further encourage them to participate.

*Proposing, deliberating, and voting*: Social activism is centrally about supporting meaningful discussion among activists, based on available information, leading to action. While simple plurality-based voting systems exist, they fail to enable the nuanced discussion, filtering, consideration, and consensus-building that is fundamental to both social activism and democratic citizenship. The *Assembly* must thus facilitate proposal-making and versioning in a legible and inclusive way - that is, for example, it should be easy to track the various versions of proposals and select among them through democratic deliberation and voting systems that encourage consensus.

*Acting Collectively*: Social actions may take different forms, depending on the purpose of the activist group. Still, activists may benefit from digital support that assist in the coordinated realization of the target actions. The *Assembly* is thus called upon to implement choreographies among participants.

*Communicating*: Communication is orthogonal and underlying all the above functions, in that activists should be provided

This work was supported in part by Inria@Silicon Valley, Inria CityLab, and the Social Apps Lab at CITRIS, University of California, Berkeley.

<sup>1</sup><https://citylab.inria.fr/appcivist-social-apps/>

<sup>2</sup><http://citriss-uc.org/initiatives/social-apps-lab/>

<sup>3</sup><http://www.denguetorpedo.com/>

with proper means to communicate with each other and with a wider population regarding both the transport and the presentation of their communication. Thus, the implementation of an *Assembly* depends on supporting middleware technology that provides the relevant connectors to compose the above functions in ways that guarantee the required extra-functional properties such as security, cost-effectiveness, and even usability.

From a software engineering perspective, architectural principles like service-orientation, coupled with the latest advances in distributed computing, appear as significant base building blocks for *Assemblies*, which can be regarded as systems-of-systems [3]. However, composing *Assemblies* through the reuse of relevant Internet-based third party services raises several challenges for the supporting software platform, most notably related to meeting the extra-functional requirements associated with social activism.

Composing systems while meeting extra-functional properties, aka QoS, has been largely addressed by software present in the enterprise context. However, the social activist context is markedly different since: *i*) activist organizations are mostly non-profit and highly sensitive to cost, *ii*) although inclusion is a very high priority, activist groups cannot afford to provide smartphones and data plans to all users, *iii*) social organizations do not have dedicated IT teams to configure and maintain the systems, and *iv*) some social activists have a very strong requirement of security and anonymity, sometimes against state actors, as has been shown by the use of Tor by activists and whistleblowers<sup>4</sup>.

The next section more specifically positions the AppCivist initiative in the broad context of Internet-enabled communication. A discussion follows of the supporting service-oriented architecture that is dedicated to enabling the composition of *Assemblies* in terms of both design- and run-time. Section IV then illustrates the unique features of AppCivist and discusses two use cases, including Dengue Torpedo, whose initial implementation informed the platform design; Section V concludes.

## II. RELATED WORK

There are several classes of Internet services available, and, indeed, already being used by activists to provide the functionalities we introduced above.

**Social Networking:** Since a large part of collective actions is communication among the group, activists have traditionally used the communication means of the day to get the word out, have discussions, plan events etc. In the digital world, this has traditionally included phone calls, SMS, and newsgroups and mailing lists. With the advent of *online social networking* (OSN), the use of sites such as Facebook and Twitter has gained momentum among activists, which leverage their abilities to rapidly reach out and discuss with a potentially large community of supporters. However, the two major concerns in using OSNs are *i*) their silo-like nature severely restricts the inclusiveness of the movement, often excluding those who

<sup>4</sup><https://www.torproject.org/about/torusers.html.en#activists>

are either too poor to access the internet regularly, or do not like the privacy implications of these OSNs; and *ii*) the limits on extensibility of the core platform makes it very difficult to customize for the functions highlighted in Section I.

**Content-management Systems:** The next level of online tools available to activists are web-based user- and content-management systems that are used by organizations; foremost among those are Google Apps for Non Profits<sup>5</sup> and Wordpress<sup>6</sup>. Each of them comes with a suite of essential services, either provided as-a-service (in the case of Google Apps) or for self-hosting (in the case of Wordpress), which includes user and content management. These built-in functionalities can then be extended by apps (in the case of Google) and plugins (for Wordpress) which are available for installation in a marketplace as well as directly from the developers' websites. Unfortunately, while these platforms provide good cores on which services can be built, there is no structured way of exploring this marketplace, let alone automatically choosing a composition of services from a set of requirements. The activist is left to use the text-based and tag-based search and filtering mechanisms available on the marketplace sites, and pore through the descriptions of the individual services to ensure that the final composition will match the functionalities sought. This trial-and-error method unfortunately can get very tiring, and crucially for activists, costly in terms of ICT-expert person-hours.

**Dedicated Services:** Another, often complementary approach involves activists hiring developers to create an in-house system to cater to their ICT needs. Fortunately, there are several excellent services catering to specific functionalities, some of which we present in the table below.

Functionality	Services
Mobilizing	CrowdVoice <sup>7</sup>
Deliberating	Liquid Feedback <sup>8</sup> , Assembl <sup>9</sup>
Acting Collectively	Crowdmap <sup>10</sup>
Communicating	Riseup <sup>11</sup> , SMSSync <sup>12</sup> , Off the Record Messaging <sup>13</sup> , Pidder <sup>14</sup> .

Unfortunately, though many of the above tools are exposed as integrable services using RESTful interfaces, the description is not in an automatically composable form, let alone one that uses a consistent taxonomy. Therefore, developers have to resort to online guides<sup>15</sup> to discover such tools, and their own technical skills to read the API documentation in order to create assemblies.

**Enterprise Platforms:** For large corporations, the service-oriented architecture is fully realized and available in the form of systems like *Enterprise Service Buses* such as IBM Integration Bus<sup>16</sup>. These come ready with advanced capabilities for service monitoring and SLA compliance in addition to the core SOA functions of service registration, discovery, and access.

<sup>5</sup><http://www.google.fr/nonprofits/products/>

<sup>6</sup><http://www.wordpress.org>

<sup>15</sup>Tools for activism: <http://en.flossmanuals.net/tech-tools-for-activism/>

<sup>16</sup>[www.ibm.com/software/products/en/ibm-integration-bus](http://www.ibm.com/software/products/en/ibm-integration-bus)

Unfortunately, although there are several open source ESBs available, using and maintaining them still remains a tedious endeavor requiring high level of skills, thus remaining out of the reach of average activists. In addition, relevant activity-enabling services need to be connected to the bus.

### III. APPCIVIST ARCHITECTURE

The AppCivist platform's architecture (illustrated in Figure 1) leverages the service-oriented paradigm, while customizing its constituents toward social activism:

**Activism Ontology.** This ontology contains the core concepts related to the functionalities needed by activists. These include concepts related to *i)* high level functions, aka *affordances*, provided by the services, and *ii)* extra-functional properties such as anonymity, inclusiveness, cost, etc.

**Activism Enabling Service Registry.** Service developers can use the AppCivist ontology to provide metadata about their services, and register them within this component. The AppCivist registry can then be used for semantic service discovery [4], [5] as needed. Note that these can easily include custom mobile apps created by the activists themselves.

**Assembly Composition Engine.** This component takes as input an "assembly" created by the activist (much like a BPMN specification in the enterprise context), in addition to the activists requirements regarding cost, anonymity, etc., and consults the AppCivist Registry to select the appropriate services that together can provide the needed composite functionality while satisfying the non-functional constraints. Note that as discussed in works such as [6], the problem of guaranteeing adequate levels of overall privacy while composing services is not completely solved.

**Assembly Execution Engine.** The execution engine actually executes the composition determined above, either as an orchestration or a choreography, as appropriate for the context. The execution engine also provides adequate communication channels between the various users and services of the service composition in question, in a manner aligned with the requirements of privacy and security, for example.

We note that although the above architecture is very similar to that of service-oriented middleware [7], [8], there are several challenges that arise due to the social activism context. Among others, due to the lack of control over the underlying network (unlike in an enterprise), the need for security, and in some cases, anonymity, is very strong. Indeed, we are exploring the possible use of open source ESBs such as Apache ServiceMix<sup>17</sup> and Red Hat JBoss Fuse ESB<sup>18</sup> as a starting point for the AppCivist platform while revisiting their core functions to match the requirements of social activism.

As discussed earlier, there are mature tools for activists already available in the public domain, albeit without a service description, let alone one based on the AppCivist ontology. We hope that the the developers of these tools, such as

Tor, would embrace the AppCivist vision and provide this metadata, which will be kept up-to-date with the evolution of the services. Given that many of these tools are open source, we intent to bootstrap the description and annotation process by contributing directly to the open source projects.

Finally, by developing services from scratch where needed, the AppCivist Service Registry will include algorithms and protocols for filtering and voting, data visualization, real-time multi-platform communication between activists and among the general public, mobile-based crowd-sourcing/sensing, and dynamic task-allocation and management among participants. Further, the interplay between mobile and cloud computing entails assessing both cost and social effectiveness. We discuss the details of our next steps in Section V.

### IV. USE CASES

Although we foresee the platform as valuable in a variety of social activism scenarios, we discuss two of them below in greater detail vis-a-vis the categories introduced in Section I to illustrate the need and applicability of AppCivist.

#### A. The Starting Point: Dengue Torpedo

The *Dengue Torpedo* project ([www.denguetorpedo.com](http://www.denguetorpedo.com)) is an ongoing social initiative in Rio de Janeiro, Brazil, Tepalcingo, Mexico, and Managua, Nicaragua, for using crowdsourcing and local involvement to track and control the mosquito vectors causing dengue fever. This is an initiative of the Social Apps Lab at UC Berkeley, whose researchers are actively working with local activists in these regions. The ICT functionalities are currently developed in-house, and are envisioned to leverage the AppCivist platform as it matures. Its current and near future functionalities for include:

- *Acting Collectively by Collaborative Mapping*, where users can indicate the location of dengue-related issues, such as stagnant water. This is currently based on Google Maps, but should be extended to plug-in, for example, Open Street Map for regions where OSM data is better.
- *Mobilizing relies on Incentive-based Action*, where users take action to eradicate the above issues (e.g., cleaning up the trash) and are rewarded for their actions. These incentives may be both virtual badges and physical prizes provided by local donors.
- *Deliberating is centered around Assembly and Consensus-Making*, so the community can better decide how to proceed to prevent future Dengue cases.
- *Communicating is by way of Mobile Apps*, which can use both data and SMS networks to enable users possessing various types of phones and service plans to contribute to the common goal.

#### B. A Foreseen Assembly: Droit au Logement

*Droit au Logement* ([www.droitaulogement.org](http://www.droitaulogement.org)) is a French organization dedicated to fighting for the rights of all individuals to have a place to live. More specifically, they work towards influencing policy-makers to improve the condition of those who are homeless or live in sub-standard

<sup>17</sup><https://servicemix.apache.org/>

<sup>18</sup><http://www.redhat.com/en/technologies/jboss-middleware/fuse>

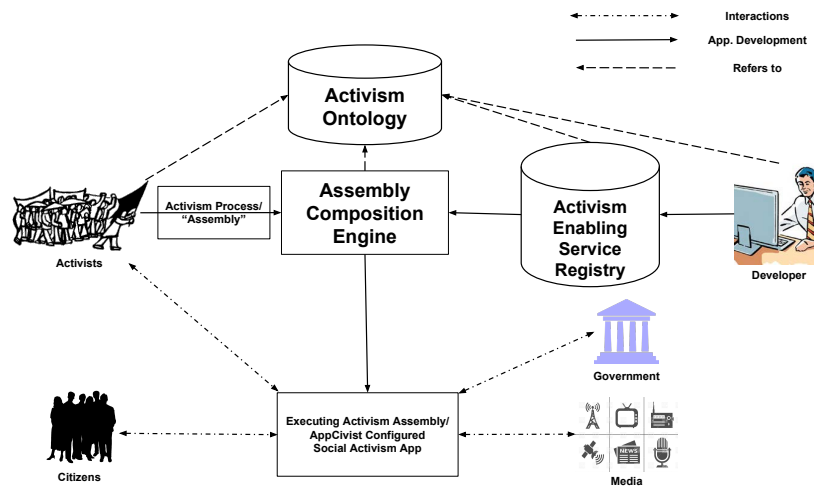


Fig. 1. AppCivist Architecture.

housing, including influencing governments, banks, and companies managing large residential properties. We envision that AppCivist assist this organization in terms of the following functionalities:

- *Mobilizing through Content Sharing*, provided through a website linked to social media, where the group can share success stories and calls to action.
- *Deliberating by way of Petition-making*, using a system that would enable activists to create and communicate petitions. Crucially, it will also enable mediapersons to track their progress, as well as the petitions' target (e.g., govt. representatives), to officially respond to them.
- *Acting Collectively by Collaborative Mapping*, where users can identify residential properties that are providing sub-standard living, or properties in good condition vacant for a long time without being rented.
- *Acting Collectively via Rapid Call-to-action*, which will be used, for example, to mobilize followers of the organization to gather quickly for a demonstration. Note that this will need to cater to those with varying technologies for communication.
- *Mobilizing based on Donation*, supporting various forms of online payment platforms including cryptocurrencies such as bitcoin, as well as in-kind donations.
- *Communicating must ensure Privacy and Security*, which is a requirement since often the work of this organization is not appreciated by the authorities, whose shortcomings the activists expose.

The above provides an initial sample of the concrete requirements that the AppCivist platform is expected to face in real-world situations. Through our existing collaborations with activists in both Paris and the Bay Area (where the Social Apps Lab is located), we intend to gather more insight about the technical requirements, as well as co-design assemblies for social activists.

## V. CONCLUDING REMARKS

With a view to improving the way social activists use ICT, the objective of our research is to study, from design to prototype implementation, a service-oriented architecture and supporting platform and services aimed at facilitating the assembly of applications by social activists. We believe that the architecture and use cases discussed in this paper are a good beginning.

In the immediate future we will *i)* Define the AppCivist ontology; *ii)* Identify, annotate, and register existing services; *iii)* Implement missing essential core services, drawing both on our expertise [9] as well as open-source tools; and *iv)* Perform studies in concert with activists both in the Paris and San Francisco Bay areas. Our user-studies will include the development of assemblies by activists working on immigrant rights, eradicating dengue fever, and aiding the homeless.

## REFERENCES

- [1] C. Shirky, *Here comes everybody: The power of organizing without organizations*. Penguin, 2008.
- [2] J. Holston, *Insurgent citizenship: Disjunctions of democracy and modernity in Brazil*. Princeton University Press, 2008.
- [3] M. W. Maier, "Architecting principles for systems-of-systems," *Systems Engineering*, vol. 1, no. 4, pp. 267–284, 1998.
- [4] D. Martin, M. Burstein, D. McDermott, S. McIlraith, M. Paolucci, K. Sycara, D. L. McGuinness, E. Sirin, and N. Srinivasan, "Bringing semantics to web services with owl-s," *World Wide Web*, 2007.
- [5] S. A. McIlraith, T. C. Son, and H. Zeng, "Semantic web services," *IEEE intelligent systems*, 2001.
- [6] R. S. Cardoso and V. Issarny, "A model for reasoning about the privacy impact of composite service execution in pervasive computing," in *Trust Management II - Proceedings of IFIPTM 2008*.
- [7] M. P. Papazoglou and W.-J. Van Den Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB journal*, vol. 16, no. 3, pp. 389–415, 2007.
- [8] V. Issarny, N. Georgantas, S. Hachem, A. Zarras, P. Vassiliadis, M. Autili, M. A. Gerosa, and A. B. Hamida, "Service-oriented middleware for the future internet: state of the art and research directions," *Journal of Internet Services and Applications*, vol. 2, no. 1, pp. 23–45, 2011.
- [9] A. Toninelli, A. Pathak, and V. Issarny, "Yarta: A Middleware for Managing Mobile Social Ecosystems," in *GPC 2011 : International Conference on Grid and Pervasive Computing*.