COMPOSE: Building Smart & Context-Aware Mobile Applications utilizing IoT Technologies

Charalampos Doukas
CREATE-NET
Trento, Italy
cdoukas@create-net.org

Fabio Antonelli
CREATE-NET
Trento, Italy
fantonelli@create-net.org

Abstract—**This paper presents an innovative framework for the development, deployment and management of smart and context-aware mobile applications. The framework utilizes Cloud computing infrastructures and Internet of Things technologies allowing the seamless integration of smart objects and external services as well as the provision of scalable resources for data and application management.**

Keywords; context awareness, smart mobile applications, internet of things

I. Introduction

The introduction of smart mobile technologies has driven information collection, presentation and user interactivity in numerous ways ([1], [2]). Typical smartphone applications have involved including gaming, productivity apps (like emails viewing, work scheduling, etc.), weather apps, travel apps, etc. Utilizing the sensors that smartphones are equipped with (such as accelerometers or location sensors), developers have been also able to built context-aware applications ([3]-[5]): Activity trackers, location-based services (like weather apps, marketing apps, etc.), recommendation applications (for tourists), smart guidance apps, etc. While smartphones provide all the technical means for developing such applications (application environment, SDKs, etc.), still the deployment of smart and context aware applications has several challenges that need to be addressed ([6]):

- Connecting to external information resources usually requires an effort for discovering the available external Application Programming Interfaces (APIs) and integrating them programmatically to the mobile application (i.e. building the interface that consumes the API and converting data for presentation).
- Building and deploying the back-end service: Usually mobile applications require a back-end infrastructure for hosting application(s) that manage user data, process and deliver content to the users.
- Maintaining the application and the data resources: Like every application, mobile applications require maintenance, especially the back-end infrastructure part (upgrading and adding more resources when needed), performing regular database and data backups, etc.

In addition, when it comes to connecting services with smart objects connected to the Internet (Internet of Things), the implementation becomes more challenging: interfaces for connecting with the smart objects are needed that overcome the heterogeneity of real-world connected things, connection issues (such as firewalls), provide bi-directional access and take into account the limited resources of IoT devices.

This work utilizes the COMPOSE (Collaborative Open Market to Place Objects at your SErvice) [7] framework for addressing the aforementioned challenges. COMPOSE provides an open-source infrastructure and a set of tools and methods for building smart applications that can communicate with smart objects (smartphones, sensors, actuators) and external information resources. The key features of COMPOSE can be summarized into the following:

- Scalable, cloud-based infrastructure featuring Platform As a Service (PaaS) for hosting back-end applications and an IoT Marketplace.
- Provision of a set of tools (SDKs, IDE, recommendation engine, etc.) for developing smart applications that can communicate with external resources.
- Provision and integration of sensor communication technologies (Web-based bi-directional communication featuring advanced Web 2.0 technologies like Web Sockets).

The rest of the paper is structured as follows: Section 2 presents background information related to context-aware applications, internet of things applications and presents related work. Section 3 discusses the COMPOSE framework while Section 4 focuses on the specific tools and methods for developing and deploying context-aware mobile applications. Section 5 presents the concept a smart, context-aware mobile application for sport content delivery that utilizes the COMPOSE framework and finally Section 6 concludes the paper.

This research has been partially supported by the EU under the ICT Collaborative Project COMPOSE (Collaborative Open Market to Place Objects at your SErvice), grant agreement n. FP7- 317862.

II. Background Information & Related Work

This section provides a brief overview of the notion of context-awareness, IoT applications and challenged and discusses some of the related work in this context.

A. Context-Aware Applications

Context awareness refers to the capability of the computing or networking applications to be aware of the existence and characteristics of the user's activities and environments. The term context-awareness in ubiquitous computing was introduced by Schilit [10], [11]. Context aware devices may also try to make assumptions about the user's current situation. Dey defines context as "any information that can be used to characterize the situation of entities." [12]

In rapidly changing scenarios, such as the ones considered in the fields of mobile, pervasive, or ubiquitous computing, systems have to adapt their behavior based on the current conditions and the dynamicity of the environment they are immersed in. A system is context-aware if it can extract, interpret and use context information and adapt its functionality to the current context of use. The challenge for such systems lies in the complexity of capturing, representing and processing contextual data. The way context-aware applications make use of context can be categorized into the three following classes: presenting information and services, executing a service, and tagging captured data. Presenting information and services refers to applications that either present context information to the user, or use context to propose appropriate selections of actions to the user. Automatically executing a service describes applications that trigger a command, or reconfigure the system on behalf of the user according to context changes. Attaching context information for later retrieval refers to applications that tag captured data with relevant context information. For mobile applications context-awareness can refer to proper content adaptation and presentation based on user location, device type and user preferences [14]. More information on context aware systems can be found in [8], [9].

B. IoT Applications & Challenges

The “Internet of Things” (IoT) is an emerging global information service architecture, which will likely be one of the most important technological advances of this century impacting a wide range of fields. IoT mainly consists of ‘things’ or smart objects that have the ability to communicate over the Internet with applications and services. The application and device management backbone needed to achieve inter-device and Internet communication can be provided by cloud computing, which facilitates scaling and provides support to billions of connected objects. In this context, the emergence of IoT can bring about increasing benefits in people’s personal and community lives. However, there exist significant inhibitors to its growth and widespread adoption with lack of interoperability being among the most prominent ones [13]. IoT refers to connecting devices with services from different vendors for the benefit of the end user. This means that developers need to develop both communication interfaces for the different hardware and the software services. Security, user and data control as well as privacy are also key IoT challenges that need to be addressed before IoT solutions can be successfully commercialized and deployed.

C. Related Work

There is a great number of works that deal with context-awareness in pervasive and mobile applications ([3],[4],[8]-[12], [14], [15]). The majority deals with utilizing contextual information about the users and their environment and utilizing the latter to adapt the content that is to presented or exchanged between the devices and remote services ([8], [9]). Appropriate selection of the available content resources can be also performed ([15]). To our best knowledge there is no complete framework that allows users to build end-to-end context-aware applications for mobile devices.

Regarding commercial or open source frameworks for Cloud-based deployment of Internet of Things applications and services, there is also a number of platforms available. Xively [25], Nimbits [28] ThingSpeak [26], Evrythng [29] and iDigi [27] are a few that could be mentioned. Xively has been one of the first on-line database service providers that allow developers to connect sensor data to the Web. Nimbits is a data processing service you can use to record and share sensor data on the cloud. It is a free, social and open source platform for the Internet of Things based on Google App Engine. ThingSpeak is another open source “Internet of Things” application and API to store and retrieve data from things using HTTP over the Internet. Evrythng focuses on making products smart by web-enabling them. iDigi Platform is a machine-to-machine (M2M) platform-as-a-service. iDigi Platform lowers the barriers to building secure, scalable, cost-effective solutions that seamlessly tie together enterprise applications and device assets. All the aforementioned platforms provide back-end support for storing, managing and visualizing sensor data. Users that wish to develop their own applications have to combine such services and use additional Cloud resources for hosting their mobile or web applications and services.

COMPOSE enables the end-to-end development and deployment of context-awareness by providing tools for collection of contextual information on smart devices, tools for communication with external resources, an infrastructure for hosting the data storage and processing, and an open and scalable marketplace, an environment where not only applications but also services derived by interacting with objects can be shared and traded. The next section discusses the framework components.

III. The Proposed Framework

The logical architecture of the COMPOSE platform is depicted in Fig. 1. The main components of the framework are the COMPOSE Marketplace, the Run-Time engine and the Ingestion layer consisting of Smart Objects and services.

The COMPOSE marketplace implements a Service-Oriented Architecture, where any resource is provided and consumed in the form of a service. An Object is then elicited to a service object when it becomes accessible through a network connection. While an object would be the sensing device monitoring the status of a house, for example, its corresponding service object is the abstraction of a given feature provided, such as data on the temperature inside the house. Service objects will comply to the COMPOSE standardized interfaces, and will be potentially running the COMPOSE runtime environment in order to be (i) accessed from the Marketplace for gathering information (ii) actuated (iii) dynamically reprogrammed at run-time. Different interfaces will be defined in order to address objects heterogeneity. Service objects can be stand-alone or composite. Composite service objects are the aggregation or composition of simple ones. For example, the house service object is the aggregation of various objects providing information on temperature, presence, light, sound, and more. Composite service objects can provide information obtained from the aggregation of multiple data flows coming from different stand-alone service objects.

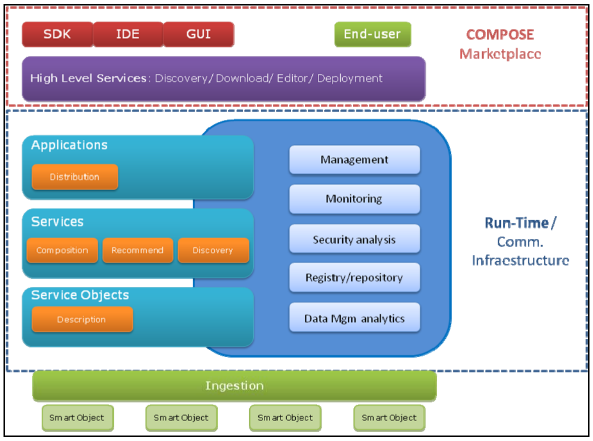


Fig. 1. COMPOSE Main Components

An object is any real-word active device capable of either providing contextual data or acting on the external environment. This includes sophisticated devices such as smartphones and multi-sensing platforms, but also simple ones like RFID tags and QR codes.

A service can be both a consumer of information originating from service objects and an actuator connected to one or multiple service object(s). When acting as a consumer, a service uses the information originating from one or more service objects to perform a given task. In contrast, when actuating on a service object, a service issues a task to such a service object. For example, a service could first read (consume) the information from a light sensor in a house, and then determine whether to switch off (actuate) the light. Services can be simple or composite. Composite services incorporate the functionalities of other services and rely on them to properly function.

End users are the consumers of the services managed through the open marketplace. A user can be a person, accessing the marketplace through the installation of a given application on a personal device or computer, or a machine, through an appropriate machine-to-machine (M2M) protocol interface, interacting with the market to integrate IoT services into its business process.

The COMPOSE open marketplace is the distributed infrastructure orchestrating all aspects of the components mentioned above. Data coming from objects can be streamed into the marketplace, where their counterpart service objects will operate, and can be published such that other entities will be able to find the information and consume it. The marketplace will ensure that privacy and security aspects are well taken care of and additional non-functional requirements such as QoS may be specified.

A developer in order to build and deploy IoT applications on the COMPOSE architecture needs to: a) Use the IDE and the SDKs and the high level services to discover existing service components or build new ones. The service components can be sets of classes or high level scripts that define what internal storage services will be used, how data will processed and stored within the infrastructures. b) Define and implement the communication with smart objects (like sensors and/or smartphones) and external resources. c) Deploy the services into COMPOSE infrastructure. The process is quite similar to deploying applications in cloud PaaS environments. More details on the COMPOSE tools and methods follow in the next section.

IV. Tools And Methods

To enable the development of smart mobile applications that can easily communicate with integrated sensors, but also with external resources, the presented framework provides specific tools, namely the Mobile Application Framework, the IoT Technologies and the COMPOSE Services.

A. The Mobile Application Framework

Specifically for the development of smartphone applications that enable context-awareness, COMPOSE has envisioned the development of a mobile application framework that enables developers to easily build apps that utilize integrated smartphone sensors and communication with the COMPOSE back-end using Web2.0 mechanisms.

Regarding context-awareness, the COMPOSE mobile framework utilizes information from the smartphone integrated sensors: the GPS and the accelerometers. The former provide location estimates of the user and communicate with COMPOSE services (discussed in following section) to retrieve location-relevant content (such as weather updates, location of friends nearby, content related to facilities nearby, etc.). The accelerometer sensors are utilized to provide an estimate of user activity. The activity levels can be used in a number of different applications (sport, quantified self applications, etc.).

In order to address the diversity in mobile development environments (Android, iOS, Windows Mobile, etc.), the mobile application framework is based on Titanium. The latter is a high level programming environment that allows developers to code once and deliver executable mobile apps for every existing environment. Thus, the framework consists mainly of Titanium-compatible Javascript classes that implement the collection of sensor information and the communication with the COMPOSE services.

B. *The IoT Communication Technologies*

Interaction with smart objects and remote services requires the utilization of IoT technologies. Smart objects do not feature only sensing devices but can also integrate various actuators (switches, motors, relay circuits) that need to be communicated by the external services. Thus, bi-directional communication mechanisms are needed that take into account the limited resources of smart objects (low cost, low power hardware, etc.) and can also be deployed behind network firewalls, NATs, etc. For this purpose, COMPOSE is adopting the Web of Things notion [24]. Each smart object is considered as a web-enabled object that can communicate over HTTP and consume REST web services. For bi-directionality, Web Sockets offer the ability to back-end services to send notification to clients (i.e. connected web objects) when needed. Clients do not need to continuously poll the servers for updates, neither to be reachable by Web (i.e. open to HTTP connections) that could be in many cases (private networks or networks over 3G) not feasible.

C. *The COMPOSE Services*

The COMPOSE Services are the software components that allow the execution of back-end applications on the COMPOSE infrastructure. The infrastructure is hosted on a Cloud-based scalable environment based on Openstack [16] and Cloudfoundry [17]. Developers can port their own applications to the proposed infrastructure by using a number of different programming languages (Java, PHP, Ruby, Node.js, etc.) and libraries for data storage and communication.

D. *The Back-End Technologies*

The framework is mainly built on top of existing state-of-the-art technologies. In particular, in the scope of the Objects as a Service work, the following technologies are being integrated:

- REST (Representational State Transfer) is the architectural principle that lies at the heart of the Web, and uses HTTP to provide application level transport.
- CouchBase Server [18] integrates an in-memory key/value store and a NoSQL back-end (CouchDB [18]) to provide a novel approach to the field of horizontally scalable databases.
- The two dominating distributed stream processing frameworks, Apache S4 [20] and BlackType Storm [21] are key technologies in the framework. They are leveraged to perform the automatic translation between service-defined data management primitives into stream processing graphs.
- Currently different options to define DSLs are being explored, from more static and simple solutions such as Apache Pig [22], to more complex approaches such as the Scala framework [23], which provides language virtualization.

V. Smart Context-Aware Content Delivery for Outdoor Sport Applications

This section describes a specific use case that has been realized for illustrating the usability of COMPOSE framework in the context of IoT smart application development. The main idea behind the use case is to demonstrate context-aware content and information delivery for outdoor sports and more specifically for skiing.

The smartphone application consists of the following components:

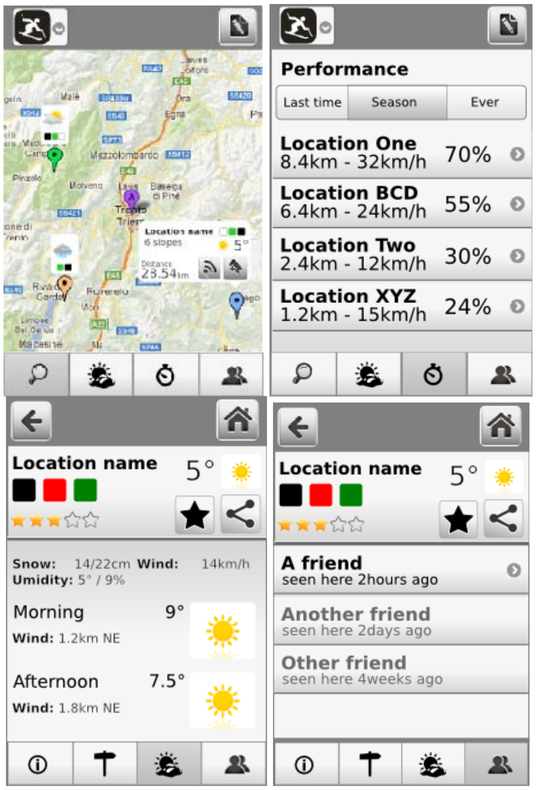
- The UI: The main graphical interface that the user is using to interact with the application, i.e. to configure user preferences (e.g., type of sport, difficulty levels, notification frequency, etc.), and receive contextual information (i.e., location of ski slopes nearby that meet user criteria, weather and social network updates based on location and user activity).
- The sensor services: These are services that run in the background and collect information about the user context. Location and activity tracking are monitored through the services and based on developer-defined conditions transmit sensor readings to COMPOSE services in order to receive updates and notifications. This service also features QR code identification. When user points the smartphone camera on a QR code, the latter is recognized and the app connects to the back-end for interpreting the QR code properly.
- The communication mechanism: A set of classes that allow developers to easily communicate with COMPOSE back-end using HTTP requests and Web Sockets for bi-directional communication.

For the realization of this particular use case, the following COMPOSE composite services have been developed and hosted by the COMPOSE infrastructure:

- The Location update service: This component receives information from smartphones about user location updates and sends back a list with nearby meteo stations, as well as ski slopes that match user criteria. In addition, it can receive QR codes from the user smartphone. Such codes can be utilised to verify user has entered specific spots (e.g., a restaurant) and redeem special coupons, etc.
- The Weather update service: This service provides access to information from all nearby weather stations and locations where current weather and forecast information is available. When there is major weather change, the service is also capable of sending alerts to clients (mobile applications) that are registered in the nearby locations.
- The Social update service: This service registers a social network of friends of the user and based on the location update service sends notifications to users about their friends being in the nearby slopes and skiing locations.

- The Activity service: The activity service collects information about the user activity levels. The latter are stored into the framework database for future reference and presentation to the user (See Fig. 2). In addition, rules can be applied and user can be notified in case low or high activity is detected..

Users upon initiating the app (which can be delivered for any available smartphone operating system) are asked to define their preferences about slope types (difficulty and length), snow types (soft, hard) and frequency of notifications. When user arrives close to a ski area, notifications are displayed for the slopes that meet the desired criteria. User can view detailed information about the slopes and also check the local weather conditions as well as a short forecast about wind, temperature, precipitation and snow fall. Upon request user can view nearby friends that have visited slopes in close distance, or automatically alerted when friends are in close proximity. Using the activity level monitoring, users can check how active they are on which slopes and under which conditions, allowing them to improve and keep track of their performance. Fig. 2 presents screenshots from the smartphone application featuring slope recommendation, activity tracker, weather and friend notification based on user location.



Location name
6 slopes
5°
Distance
23.54km

Performance

Last time | Season | Ever

Location One
8.4km - 32km/h 70%

Location BCD
6.4km - 24km/h 55%

Location Two
2.4km - 12km/h 30%

Location XYZ
1.2km - 15km/h 24%

Location name 5°
⬛ 🟥 🟩
★★★☆☆

Snow: 14/22cm Wind: 14km/h
Umidity: 5° / 9%

Morning 9°
Wind: 1.2km NE

Afternoon 7.5°
Wind: 1.8km NE

Location name 5°
⬛ 🟥 🟩
★★★☆☆

A friend
seen here 2hours ago

Another friend
seen here 2days ago

Other friend
seen here 4weeks ago

Fig. 2. Smarphone app screenshots illustrating the slope recommendation, activity trackter, weather and friend notification based on user location

VI. Conclusions

COMPOSE projects provides among others tools and an infrastructure for developing and hosting smart and context-aware mobile applications. Communication with sensors and external resources provides customized information and appropriate content adaptation based on context and user preferences. The COMPOSE Marketplace serves as a back-end infrastructure for hosting the smart services that process and store user data. Future work includes the evaluation of the proposed framework and the developed use case in a real environment. In addition, the enhancement and further development of COMPOSE platform will allow to integrate better security and user/device authentication mechanisms, as well as service discovery and recommendation services for building more content-rich applications.

Acknowledgment

This work has been partially supported by the EU under the ICT Collaborative Project COMPOSE (Collaborative Open Market to Place Objects at your SErvice), grant agreement n. FP7- 317862.

References

[1] Antoniou, Z.; Varadan, S., "Intuitive Mobile User Interaction in Smart Spaces via NFC-Enhanced Devices," Wireless and Mobile Communications, 2007. ICWMC '07. Third International Conference on , vol., no., pp.86,86, 4-9 March 2007

[2] Razzak, F.; Bonino, D.; Corno, F., "Mobile interaction with smart environments through linked data," Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on , vol., no., pp.2922,2929, 10-13 Oct. 2010

[3] Lowe, R.; Mandl, P.; Weber, M., "Context Directory: A context-aware service for mobile context-aware computing applications by the example of Google Android," Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on , vol., no., pp.76,81, 19-23 March 2012

[4] Buthpitiya, S.; Luqman, F.; Griss, M.; Bo Xing; Dey, A.K., "Hermes -- A Context-Aware Application Development Framework and Toolkit for the Mobile Environment," Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on , vol., no., pp.663,670, 26-29 March 2012

[5] Gehlen, Guido; Mavromatis, Georgios, "Mobile Web Services based Middleware for Context-Aware Applications," Wireless Conference 2005 - Next Generation Wireless and Mobile Communications and Services (European Wireless), 11th European , vol., no., pp.1,7, 10-13 April 2005

[6] Jason I. Hong and James A. Landay. 2001. An infrastructure approach to context-aware computing. Hum.-Comput. Interact. 16, 2 (December 2001), 287-303.

[7] COMPOSE Project, FP7- 317862, http://www.compose-project.eu/

[8] Charalampos Doukas and Ilias Maglogiannis, “Intelligent Pervasive Healthcare Systems”, in Advanced Computational Intelligence Paradigms in Healthcare – 3 Series: Studies in Computational Intelligence, Springer Eds. Vol. 107, 2008, pp. 95-115, Hardcover, ISBN: 978-3-540-77661-1

[9] Charalampos Doukas, Ilias Maglogiannis, Kostas Karpouzis, “Context-Aware Medical Content Adaptation through Semantic Representation and Rules Evaluation”, presented at 3rd International Workshop on

Semantic Media Adaptation and Personalization – SMAP2008, December 15-16, Prague, Czech Republic.

[10] B. Schilit, N. Adams, and R. Want., "Context-aware computing applications". IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), Santa Cruz, CA, US: 89-101, 1994.

[11] Schilit, B.N. and Theimer, M.M., "Disseminating Active Map Information to Mobile Hosts". IEEE Network 8 (5): 22–32., 1994

[12] A. Dey, Anind K., "Understanding and Using Context". Personal Ubiquitous Computing 5 (1): 4–7, 2001

[13] Yen-Kuang Chen, "Challenges and opportunities of internet of things," Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific , vol., no., pp.383,388, Jan. 30 2012-Feb. 2 2012

[14] Hofer, T.; Schwinger, W.; Pichler, M.; Leonhartsberger, G.; Altmann, J.; Retschitzegger, W., "Context-awareness on mobile devices - the hydrogen approach," System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on , vol., no., pp.10 pp.,, 6-9 Jan. 2003

[15] Pawar, P.; Van Halteren, A.; Sheikh, K., "Enabling Context-Aware Computing for the Nomadic Mobile User: A Service Oriented and Quality Driven Approach," Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE , vol., no., pp.2529,2534, 11-15 March 2007

[16] Openstack, Open source software for Cloud services, http://www.openstack.org

[17] Cloudfoundry PaaS, http://www.cloudfoundry.com/

[18] CouchBase Server. http://www.couchbase.com

[19] Apache CouchDB. http://couchdb.apache.org

[20] S4: Distributed Stream Computing Platform. 2010 IEEE International Conference on Data Mining Workshops (ICDMW). Dec. 2010. Neumeyer, L., Robbins, B. ; Nair, A. ; Kesari, A. pp 170-177.

[21] STORM: Distributed and fault-tolerant realtime computation. http://storm-project.net/

[22] Apache Pig. http://pig.apache.org/

[23] Scala Framework. http://www.scala-lang.org/node/273

[24] Dominique Guinard, Vlad Trifa, Friedemann Mattern, Erik Wilde, *From the Internet of Things to the Web of Things: Resource Oriented Architecture and Best Practices*.In: Dieter Uckelmann, Mark Harrison, Florian Michahelles (Eds.): Architecting the Internet of Things. Springer, ISBN 978-3-642-19156-5, pp. 97-129, New York Dordrecht Heidelberg London, 2011

[25] The Xively Cloud Service, http://www.xively.com

[26] Internet of Things – ThingSpeak service, http://www.thingspeak.com

[27] iDigi Device Cloud, http://www.idigi.com

[28] Nimbits Data Logging Cloud Sever, http://www.nimbits.com

[29] Evrythng, http://www.evrythng.com/