FUTURE PERSPECTIVE ON THE EXPLOITATION OF REMOTE SENSING DATA

Timo Bretschneider
Airbus Group Innovations Singapore
110 Seletar Aerospace View, Singapore 797562, Republic of Singapore
Email: timo.bretschneider@airbus.com

KEY WORDS: Cloud computing, third-party modules, pay-by-use

ABSTRACT: This paper describes a system in which satellite image data is processed "blindly" and ubiquitously hosted on a computing architecture, i.e. Cloud, operated by the data provider. In particular, users pay by use for data access and resource utilisation without actually owning the data or even being able to visualise a full scene. Although this concept appears counterintuitive at first glance, it actually enables a wide range of new and cost effective applications. Much larger datasets can be used – for instance in case of temporal series – since the user is only allowed to mine the remote sensing data without visualisation of the original imagery. Hence, the price per data item can be much lower than in case of a licencing / ownership model. Other examples of benefitting applications are use cases which combine data from different satellites in order to produce operationally stable results. The aspect of saving costs is obvious. In addition, by bringing the application to the data instead of bringing the data to the application, data transfer bottleneck can be avoided. However, more importantly by taking a modular approach in designing applications, users can decide to provide their encapsulated expertise in application modules to other users and financially participate in the pay-by-use model. This concept is similar in nature to Apple's iTunes, while the overall system is leveraging on trends and concepts in the consumer market like mobile access to large content archives (Amazon, Netflix), satellite imagery for everybody (Google), and utilisation of computing resources (Hewlett Packard). Application examples are discussed and analysed.

1. INTRODUCTION

The continuous improvement of higher spatial, spectral and temporal resolutions drove the development of increasingly powerful applications for GIS, Intelligence etc. with the governmental sector contributing the main part of the business with additional revenues from high value industries like oil and mining. Other sectors like farming can greatly benefit from information contained in remotely sensed imagery, e.g. for precision farming, but never managed to establish larger footprints outside their niche markets. The main reason is the relative high cost for the data, which often renders the use of remote sensing imagery uneconomical even though the strong market pressure let to a downwards trend of the price per unit of data in the past two decades (Satellite Industry Association, 2015). Nevertheless, the market size is relative constant with all players facing increasing pressure from missions like the European Sentinels satellites providing *de facto* free-of-charge data for a variety of users and applications.

At the same time the data distribution itself has not developed much besides the use of storage media with higher data densities and the incorporation of the Internet for browsing and product delivery. Last but not least, the enhancements in resolutions let to data volumes in the range of several gigabytes and the terabyte boundary for everyday applications is not too far away – data amounts that cannot easily be transferred over broadband Internet. In summary, while the current business model in the remote sensing industry can address a wide range of customers successfully, it is experiencing (and will experience even more) financial and technical difficulties in the future.

The most common sales model for satellite imagery is depicted in Figure 1a. The data is sold to the customer who applies own intellectual property (IP) using own hardware. This model requires a highly skilled customer and generally can be found in the context of government agencies and specialised service providers. Alternatively, also value-added products can be offered which are generally created on the satellite provider's side. This approach, which is shown in Figure 1b, often can be found if specific IP is not available to the customer. Figure 1c displays a very similar case, however, the actual image data is provided by the customer. Extending the previous two cases, Figure 1d shows the situation where IP and specialised hardware infrastructure is bundled as a product, e.g. the Pixel Factory (Airbus Defence and Space, 2016), for users who then can use either foreign or own data for processing. In all the previous scenarios it was assumed that the data delivery channel from the data provider to the customer is sufficient. However, in many instances communication channels cannot keep up with the rapidly increasing data volume, i.e. if (near) real-time data access is required. Hence, this paper proposes the concept illustrated in Figure 1e. Instead of sending remotely sensed data to the customer, the satellite imagery remains with the provider and customers can directly use the provider's IP and the available processing platform. The advantages of this approach are twofold: Firstly, users can build upon trusted IP and do not need to develop all technologies from scratch. The access to data and processing IP is not limited to the services made available by the provider, but are open to third parties as well. This value preposition of an open access is one of the central aspects of this paper and is explained in detail in the next section. Secondly, the entire data archive can be accessed and applications are not restricted to individual scenes opening new opportunities previously unknown.

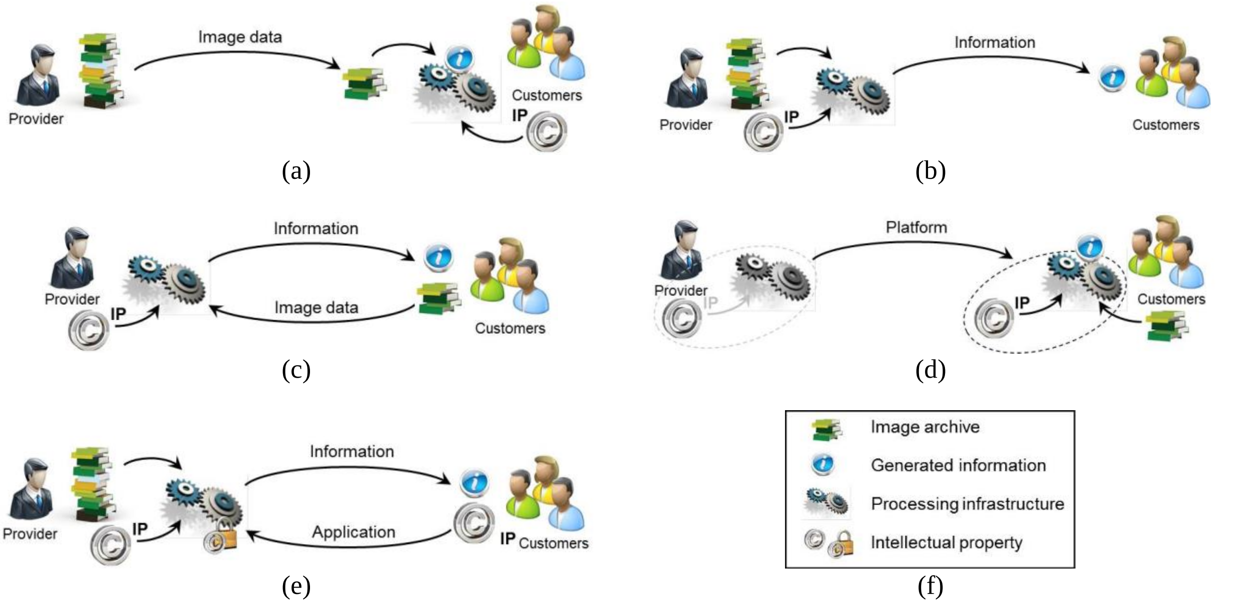


Figure 1: Data processing relationship models: (a) standard image sales model, (b) value-added high quality products, (c) provision of services, (d) distribution of platform solutions, (e) data centric approach, (f) legend

This paper is organised as follows: Section 2 describes the methodology of the new approach and discussed one possible instance of system architecture. An example for how applications can be realised is given in Section 3. Finally, Section 4 concludes this paper.

2. METHODOLOGY AND SYSTEM ARCHITECTURE

Various concepts for system architectures are conceivable and in the following only one possibility is discussed to illustrate the general concept. As can be seen in Figure 2, raw data satellite data is collected in archives and made available to applications running in a strictly encapsulated environment on the application executor. The main objective of this execution block is to guarantee data integrity in the archives, to block any unauthorised data transfer from and to the archives as well as to keep applications from each other separated, similarly to the concept of "sandboxes" in the iOS from Apple. The actual realisation of such architecture can for instance be based on Cloud services (Gunasekera *et al.*, 2015). As far as the business case is concerned, the main feature is the cashier block which determines the charges to be paid by the user based on data usage (type, volume etc.) and utilisation of provided computing infrastructure itself (processing, memory etc.). The main difference of the proposed concept towards a simple hosting of data and functionality is that users are not enabled to visualise or extract large amount of data directly. Instead, all data can only be processed "blindly", i.e. applications can access the data and extract relevant high level information. Only this information is made available to the user of the corresponding application. This differentiation is essential in order to enable a different price model in contrast to the licensing of remotely sensed data nowadays. In summary, customers can chose between the options of obtaining satellite imagery with the focus on the highest level of quality during all processing steps or rather settling for possibly a lower level of quality using larger amount of data at a much lower price.

In order to deploy applications successfully in this environment, a so-called nursery is provided. Applications can be tested using a limited set of satellite imagery without creating costs for the data access and at the same time users can visualise the corresponding images in order to improve / enhance the processing. Once a module's performance is considered satisfactory, the module can be deployed system wide. Moreover, interfaces to established tools like ENVI, ArcGIS etc. permit the continuous use of already established customer IP as well as the support of legacy code.

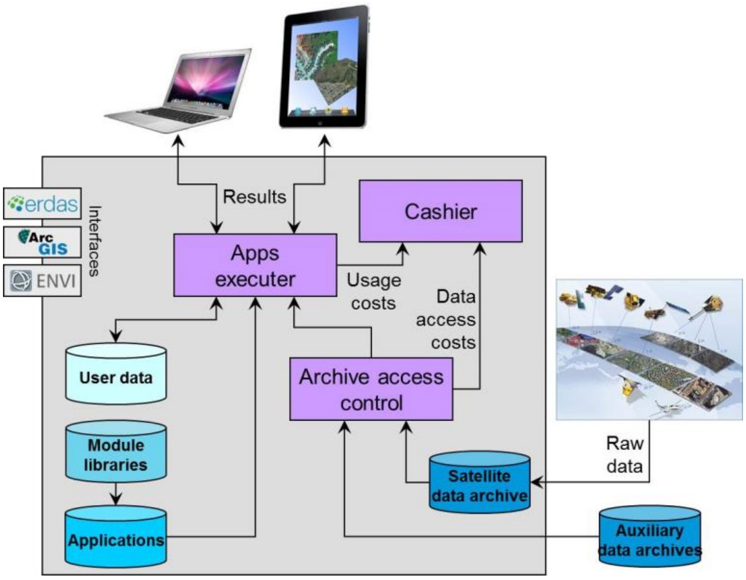


Figure 2: System architecture

The implementation of applications can follow different methodologies as well. In the following, a workflow-based approach is described as one possible solution. Modules can be linked by the user to create a flow of processing steps. The actual functionalities of modules can either be created by integrating already existing modules in workflows or writing program code.

A specific class of modules are those which can access the data archives or export (include visualisation) of the image data. In particular, these modules contain monitoring capabilities that guarantee data integrity and limit excessive data export. This aspect is indicated by the overlooking "supervisor" in **Figure 3**. The main crux in the design of so-called module templates, i.e. modules that can be used to implement the actual processing functionality, is the interfacing. Images and auxiliary data need to be standardised in order to enable the flow from on module to the next. This task is always performed before the processing and is within each module's responsibility. Hence, each module can decide which data it accepts and how this information is handled rather than imposing the requirement of standardising the processing output. The benefit is an easier integration of legacy modules in the long-run. Specific information that allows for controls is available as well and determines the behaviour within modules and among modules. Two of the most relevant controls are the ones for monitoring the data usage and utilisation of hardware resources.

As mentioned previously, the processing function inside a module can be built up from other workflows or through writing program code directly. Users can either define the resulting modules as "*private*" in case they do not want to share their IP with others or as "*public*". The latter option lets the provider of the module participate in the created revenues. For instance, if another customer runs a public module, then the creator of this module receives a percentage of the computed costs that were charged to the actual user. The philosophy behind this approach is similar to Apple's iTunes and Google's Play Store that only a vibrant landscape of third parties can fully explore the potential of the underlying system.

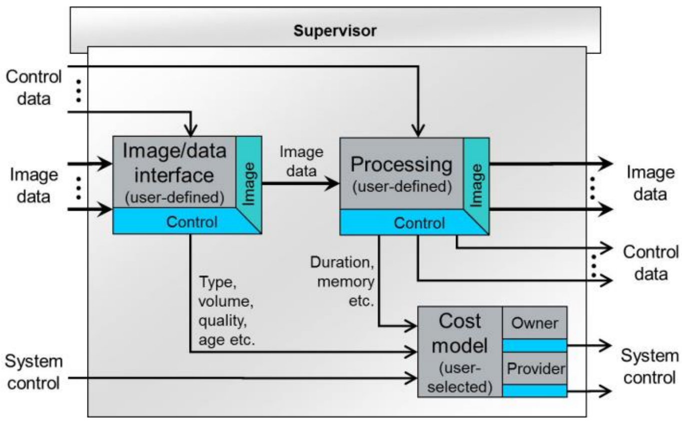
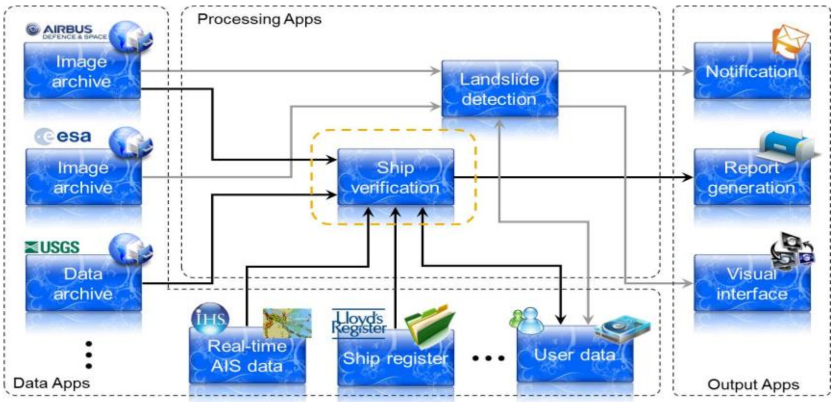


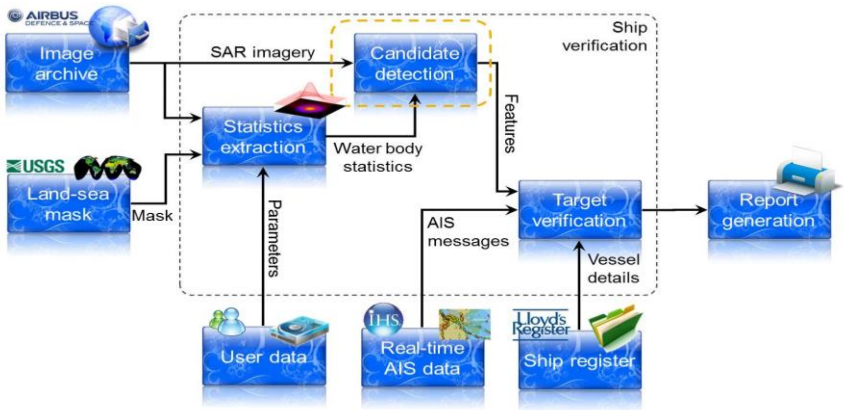
Figure 3: Generic structure of a module

3. APPLICATION EXAMPLE FOR WORKFLOW DESIGN

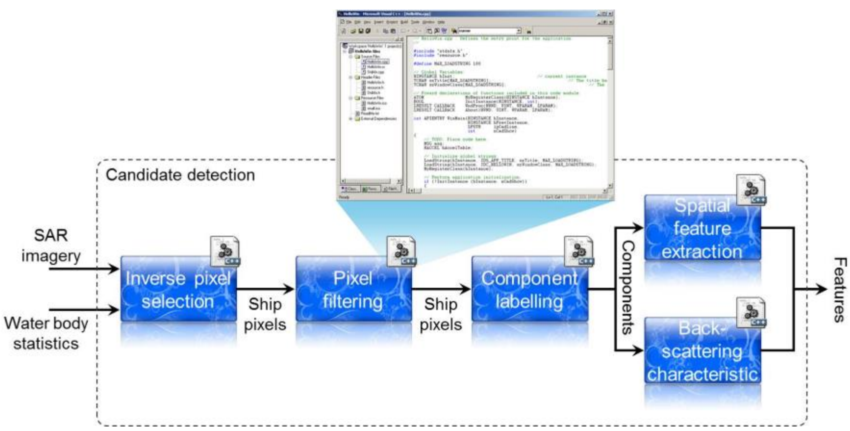
This section of the paper provides an abstract example for the workflow design of applications and shows how the various modules are refined. Other examples for workflow processing can be found in (Esri, 2016) and (Stratoulias *et al.*, 2016). Figure 4a shows two simplified workflows enabling ship verification and landslide detection, respectively, based on different sets of input data creating email notifications, reports and visualisations. Note that auxiliary data is utilised for the ship detection, i.e. real-time data from the maritime automatic identification system (AIS) provides meta-data (position, speed, heading, ship name, ship type, last port of call etc.) as supplementary information to the detected ships in the satellite imagery. Additional details can be retrieved from Lloyd's Register. Finally, a hardcopy report is created and extracted details stored as user data. For instance, the latter can be used to build up an own database of suspicious ships.



(a)



(b)



(c)

Figure 4: Application workflows: (a), top-level view with application names and main data flows, (b), break-down of application in modules, (c) bottom-level view with low-level processing modules and source code implementation

In a more detailed view in Figure 4b, the application of ship detection is broken down into individual modules. The external data flow (from the archives and to the output devices) is refined and addresses specific modules, which follow the structure introduces in Figure 3. The depicted workflow does not provide any information on the public or private nature of the modules; however, the workflow itself remains the same for both cases. The used modules in the workflow were selected from the pool of available modules, arranged by drag-and-drop and combined through the definition of the data flow. For the reason of simplicity, the control signals were omitted here. It is worthwhile mentioning that loops in workflows pose a major challenge in the context of workflow correctness analysis and, thus, should be avoided.

As described previously, modules can either comprise a refined workflow of modules (recursive implementations are not advised) or actual executable program code. An example is illustrated in Figure 4c where the candidate detection module in Figure 4b is broken down further. At one stage this refinement process ends at the bottom level (without limitation on the possible refinement depth). This is shown in Figure 4c with the module for pixel filtering containing the actual implementation. Alternative to the actual program executable file, interface modules can integrate already existing implementation in Matlab etc.

4. CONCLUSIONS

The main objective of the proposed system is to make satellite imagery available to a larger variety of application domains. In particular, areas that do not require the highest degree of quality and at the same time need to be more cost efficient can benefit from the described approach. The given examples are just a beginning and limits will be continuously pushed as it is the case for smartphones and tablet computers with daily expanding scopes of applications. In particular, mobile applications can benefit. The two cornerstones of the proposal are an open system architecture that allows the straightforward integration of third party data and functionalities as well as the participation of third parties in the revenue model.

In Figure 5 a selection of applications was related towards current image and information products in order to map the target market with respect to the delivered image quality and the level of information complexity.

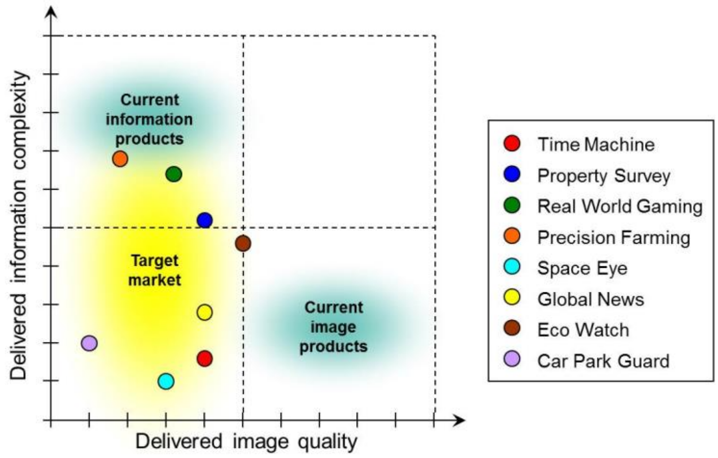


Figure 5: Application partitioning

Even if for some applications the depicted approach is not desirable, in the long-run the data volumes will be too large to be transferred in a timely manner from the provider to the customers.

REFERENCES

Airbus Defence and Space, 2016. Pixel Factory™ – The power of an industrial solution in your hands, Retrieved September 19, 2016, http://www.intelligence-airbusds.com/en/161-pixel-factory.

Esri, 2016. Image Management Workflows. Retrieved September 20, 2016, http://resources.arcgis.com/en/.

Gunasekera, K., Hunter, J., Canto, M., Weeks, S., 2015. Accelerating satellite image processing through Cloud computing. In: Proceedings of the International Conference on Computer Technology and Information Systems.

Satellite Industry Association, 2015. State of the Satellite Industry Report, Retrieved September 19, 2016, from http://www.sia.org/wp-content/uploads/2015/06/Mktg15-SSIR-2015-FINAL-Compressed.pdf.

Stratoulias, D., de By, R.A., Tolpekin, V., Zurita-Milla, R., Bijker, W., Hasan, M.A., Vermote, E., 2016. A workflow for automated satellite image processing: From raw data to object-based spectral information. In: Proceedings of the 2nd Joint Workshop of the EARSeL Special Interest Group on Land Use & Land Cover and the NASA LCLUC Program.