A Personal Analytics Platform for the Internet of Things
Implementing Kappa Architecture with Microservice-based Stream Processing

Theo Zschörnig[1], Robert Wehlitz[1] and Bogdan Franczyk[2,3]
[1]*Institute for Applied Informatics (InfAI), Leipzig University, Hainstr. 11, 04109 Leipzig, Germany*
[2]*Information Systems Institute, Leipzig University, Grimmaische Str. 12, 04109 Leipzig, Germany*
[3]*Business Informatics Institute, Wrocław University of Economics, ul. Komandorska 118-120, 53-345 Wrocław, Poland*

Keywords: Personal Analytics, Internet of Things, Kappa Architecture, Microservices, Stream Processing.

Abstract: The foundation of the Internet of Things (IoT) consists of different devices, equipped with sensors, actuators and tags. With the emergence of IoT devices and home automation, advantages from data analysis are not limited to businesses and industry anymore. Personal analytics focus on the use of data created by individuals and used by them. Current IoT analytics architectures are not designed to respond to the needs of personal analytics. In this paper, we propose a lightweight flexible analytics architecture based on the concept of the Kappa Architecture and microservices. It aims to provide an analytics platform for huge numbers of different scenarios with limited data volume and different rates in data velocity. Furthermore, the motivation for and challenges of personal analytics in the IoT are laid out and explained as well as the technological approaches we use to overcome the shortcomings of current IoT analytics architectures.

1 INTRODUCTION

It is estimated that the number of Internet of Things (IoT) devices will grow in huge quantities, to around 24 billion in the year 2020 (Greenough, 2016). This ever-increasing number of IoT devices creates vast opportunities for businesses and industry but also for common individuals (Ruckenstein, 2014). In order to gain meaningful insights, it is necessary to provide analytics platforms which are able to process, integrate and enrich the data from IoT devices. Current research in the field of IoT analytics focuses on different domains, such as health care, energy and utilities and manufacturing (Stolpe et al., 2016). Yet, in order to further enhance the usefulness of IoT devices to consumers, it seems plausible to provide powerful personal analytics.

During our research, we found that these kinds of analytics have different challenges and technological requirements compared to common IoT analytics architectures and therefore need new approaches to be handled. Against this background, we present an architectural approach for an IoT analytics platform in the context of personal analytics.

In this paper, we describe the motivation to conduct this research and the challenges when designing architectures for IoT personal analytics platforms but also the opportunities they provide (Section 2). We give an overview of the state of the art in IoT analytics regarding technologies and architectures and show that these are not fully suitable for personal analytics (Section 3).

Further, this paper presents technological approaches to resolve these issues and challenges (Section 4). The main contribution of this paper, an approach to build an analytics platform architecture which is able to be used for personal analytics, is described in Section 5. In conclusion, we provide ideas to further the research in this field (Section 6).

2 MOTIVATION AND CHALLENGES

The usage, adoption and impact of the IoT can be categorized into levels of society, industry, organization and individuals (Riggins and Wamba, 2015). With the growing number of IoT devices used in everyday life, it is necessary to gather richer insights in how to use the data not only on a high, aggregated level. A smaller, more intimate scale and use, by individuals, commonly referred to as personal analytics (Choe et al., 2014) should also be looked upon. Lacking a unanimous definition, we describe

Zschörnig, T., Wehlitz, R. and Franczyk, B.
A Personal Analytics Platform for the Internet of Things - Implementing Kappa Architecture with Microservice-based Stream Processing.
DOI: 10.5220/0006355407330738
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 733-738
ISBN: 978-989-758-248-6

personal analytics as analytics of data produced by an individual. It can also be seen as analytics of data from or linked to a specific individual. Therefore, personal analytics call for user-friendly applications which empower self-service capabilities. The types of analytics used in this regard are descriptive, predictive and prescriptive (Swan, 2012).

Since IoT topics like Smart Home and home automation have become more popular in recent years, but still struggle to gain broader acceptance (Accenture, 2016), it seems plausible to extend the field of personal analytics to these. This enables consumers, for instance, to gain insights into their own energy consumption and device usage in the closed environment of their homes. Further, the complex interaction of IoT devices as well as their smart usage can be supported by the use of machine learning, data mining, clustering and analytics insights, enhancing the usage value of them to the consumers.

Providing an analytics platform or tools to consumers is usually part of IoT platforms (Mineraud et al., 2015). They can be vendor-agnostic, third-party-based or open source and omit the need for consumers to build their own management and control systems to use their IoT devices.

In the context of providing an IoT platform for large numbers of consumers, IoT analytics platforms face several architectural challenges. Semantics of the data to be collected and analysed change frequently over time and are sometimes unknown (Xu et al., 2016). Also, the ability to save large volumes of different kinds of structured and unstructured data (Hasan et al., 2015), in a scalable, easy updatable manner is important. Furthermore, they need to process real-time data (Rozik et al., 2016) and integrate it with historical data, extend data processing capabilities without ease and provide the gained insights to different endpoints (Cheng et al., 2015). Lastly, they need to be able to combine events of different IoT devices for meaningful information and predict events based on the data (Rozik et al., 2016).

In context of personal analytics, we found the requirements for IoT analytics platforms to be different. Major differences are data volume and velocity to be analysed. Whereas common Big Data technologies aggregate data from huge numbers of data sources thus creating large volumes of data, the number and therefore volume in personal analytics is much smaller. Corresponding architectures still need to be able to handle huge volumes and high velocity of data, but only at the infrastructure level. Instead of processing and computing a modest number of Big Data problems, the analytics architecture has to compute large numbers of smaller problems. Since every consumer is able to define their own analytics use cases, the resulting applications do not need the same computational power as common Big Data scenarios and, as a consequence, should be designed in a flexible and lightweight way.

This shift leads to huge numbers of different analytics scenarios in terms of data sources, data processing and transformation needs as well as insights gained. As a result, the architectures of the platform must be able to provide large quantities of processing and analysis algorithms which can be easily replaced in user-created analytics pipelines. Still, the already established architectural requirements for IoT analytics platforms apply.

Looking at current solutions, these requirements seem to have only been insufficiently met. Therefore, we propose a new more flexible architecture which is able to satisfy the needs of personal analytics, especially in IoT platform environments.

3 STATE OF THE ART

As mentioned before, current IoT analytics platforms research and solutions mainly employ Big Data technologies in order to tackle the architectural requirements of IoT analytics scenarios. Commonly used are Big Data processing frameworks for batch and stream processing, such as Apache Spark, Storm, Samza, and Flink, to be composed in a Lambda Architecture (Cheng et al.; Hasan et al., 2015; Rozik et al., 2016). This architectural concept includes a batch, stream and serving layer. The batch layer is used to store all ingested data as well as compute views on the data continuously. Since batch processing huge data volumes creates high latency, the speed layer is used to compensate this and create incremental real-time views of the data. The real-time views complement the batch views. This creates the need to develop two data processing logics. In addition, the development of processing algorithms using processing frameworks is rather cumbersome and has a steep learning curve for developers.

There have also been works which use Business Intelligence applications (Chang et al.; Mishra et al., 2015) to implement IoT analytics or related problem fields for companies. In addition, Complex Event Processing (CEP) is used to analyse events of IoT devices and link them to external data sources (Naqishbandi et al., 2015), but also add another level of complexity to data processing and analysis.

IoT analytics in general are object of investigation in a multitude of domains. This research, especially in energy and utilities, mainly focuses on aggregated insights of broad applications, such as smart cities (Ramakrishnan and Gaur, 2016) or smart grid (Hasan et al., 2015).

However, none of the related works consider the challenges for IoT analytics platforms which arise in the context of personal analytics. This paper aims to provide an architectural approach to fill this gap.

4 TECHNOLOGY

In this section, we describe the core technologies and technological approaches we use to implement our IoT personal analytics architecture.

4.1 Kappa Architecture

The foundation of our approach to IoT stream processing is the Kappa Architecture. It is derived from the more commonly used Lambda Architecture but tries to overcome its shortcomings. Comparing both architectures, Stolpe (2016) points out that the development of algorithms for both processing layers of the Lambda Architecture, the batch and the stream layer, is disadvantageous. Therefore, the main concept of the Kappa Architecture evolves around the idea to drop the batch layer and only use a stream processing system (Wingerath et al., 2016). In case the underlying logic changes, all historic datasets are reprocessed (Kreps, 2014; Wingerath et al., 2016) and the "old" output data tables of the serving layer are dropped (Kreps, 2014). For this to work, usually the data source is a (distributed) log data store, such as Apache Kafka. Therefore, the Kappa Architecture, in contrast to Lambda Architecture, allows for more flexible adaption of changing processing and analytics requirements since the overhead of a second processing layer is mitigated.

Providing increased flexibility and reduced overhead, the Kappa Architecture is not without trade-offs. Especially, increasing data volumes require more computational power or better data compression, thus making the Kappa Architecture only a viable approach in systems with either high computational power, finite data retention rates or sufficient data compression (Wingerath et al., 2016).

Looking at IoT data being dominantly time-series data with rapidly changing, oftentimes unknown, context and analytics concepts, the flexible and lightweight nature of the Kappa Architecture enables it to cope with the challenges these kinds of data provide.

4.2 Microservices

In recent years, the use of microservices for building flexible software architectures has become rather popular. In environments with fast changing requirements, microservice architectures offer a variety of advantages over traditional approaches. They are characterized as a set of small services, developed along business requirements and are completely independent from one another (Lewis and Fowler, 2014). They are loosely coupled and focus on a single task, and are therefore easily changeable or even replaceable (Fetzer, 2016).

The microservice paradigm is closely linked to the DevOps approach which advocates tight collaboration between software development, execution and maintenance as well as automated software delivery. Microservices are often implemented using operating-system virtualization or container engines, such as Docker (Jaramillo et al.; Ueda et al., 2016). This adds to their fast and flexible deployment and also makes them easily transferable.

4.3 Stream Processing and Libraries

Stream processing is a major concept in an IoT analytics architecture. Data is constantly emitted by IoT devices thus creating the need to constantly update and increment existing data views.

Stream processing libraries are software libraries which are used to implement data extraction with task and pipeline parallelism. To achieve this, they leverage the functional capabilities of either a programming language (RaftLib, Auto-Pipe, WaveScript) or an application system, which usually is some kind of data source (Kafka Streams) and provide these for usage in stream processing applications. The latter type of libraries is fairly new and since they do not require the setup of complex application architectures for processing jobs, they are more lightweight than the usually used Big Data processing frameworks.

After conducting a literature review, we found that there is no substantial research on how they actually compare to Big Data frameworks in terms of computational speed and parallelism, especially considering Big Data problems. Still, they are an easy to learn alternative, showing lots of potential for use in Kappa Architectures.

4.4 Data Lake

The concept of the Data Lake is often used, when it is necessary to store large amounts of data without knowing their context or later use. Therefore, it is characterized as a data store, which does not employ a specific storage technology implementation but rather a set of typically NoSQL and In-Memory databases complemented by relational databases (Pasupuleti and Purra, 2015). It stores vast amounts of structured as well as unstructured data in low cost technologies (Fang, 2015) and supports flexible data

models and caters to data scientists and data exploration instead of rigid business applications (Pasupuleti and Purra, 2015). One of the major benefits using the Data Lake concept is that it is not necessary to transform or process data before its actual use (Fang, 2015). The data in the Data Lake is supposed to be open to further investigation to all members of an organization (Fang, 2015). In order to purposefully use the data in the Lake it is necessary to build and maintain a metadata repository which enables meaningful semantic connotation of all data (Alrehamy and Walker, 2015).

5 SOLUTION PROPOSAL

Because of the limitations of existing Big Data analytics architectures in general and IoT analytics platform architectures more specifically, we designed a new architectural approach for handling personal analytics in IoT environments.

The platform architecture is based on the previously introduced concept of the Kappa Architecture and is shown in Figure 1. In the context of the IoT, data sources can be categorized into sensors, actuators and tags. They form the main data sources which are relevant to the IoT analytics platform. Still, it is possible to integrate other external data sources to provide context, such as meteorological data.

The data IoT devices emit is pushed into a log data store. Whereas it is possible to use other alternatives, we used Apache Kafka due to its rich feature set, easy integration with other used technologies and its architectural distribution capabilities. These are all features which complement the overall requirements of handling a huge number of heterogeneous data streams. The actual data ingestion and push to the log data store is achieved using IoT middleware, such as Node-RED.

While it is possible to have all data from one IoT device put into one topic in the log data store together, a lot of devices offer a variety of IoT services which in turn encapsulate different sensors or actuators. This makes it more feasible to have topics based on IoT services rather than devices. The topic is set by the control service and the identification is saved as metadata in an external device repository and is associated to an actual IoT device. This metadata repository is the basis for later reprocessing tasks as it enables the platform to identify topics which require reprocessing due to changed requirements.

The data in the log data store is processed using a lightweight stream processing system. It needs to be easily adaptable to changing data models and analytics requirements. Also, the technological overhead for implementation needs to be low, so programmers can easily be introduced to enhance, maintain and test existing or develop new processing applications. A microservice architecture is suitable to fulfil these requirements. Rather than using a full-fledged stream processing framework, each processing task is done by a single microservice. The microservices access the data directly from the log data store and transform it as needed using stream processing libraries. The control service accesses the metadata of different stream processing microservices from a processor repository. This information is used to start processing instances as needed.

To utilize the full functionality of the distributed log data store, we used Kafka Streams. The microservice stream processing system can be scaled horizontally in regard of single topics but also as a whole system. Computation intensive transformations can be scaled out by starting additional microservices using the same processing algorithm, and the system itself can be scaled out to adequately compute huge amounts of topics. The feasibility of this stream processing architecture relies heavily on the nature of IoT personal analytics which is to handle problems at a much smaller scale than in common Big Data scenarios.

Processed data is pushed back into the log data store as a new topic. At this point all data takes two different paths of further usage. Since meaningful IoT analytics applications rely heavily on near real-time data, it is only natural, that all processed data is pushed into an analytics data store which is the serving layer of the Kappa Architecture. The data store should be column-oriented or optimized for time series data. Examples of time-series databases are Graphite or InfluxDB. Using the serving layer, it is possible to access and query all processed data in a near real-time fashion. In addition, multiple application programming interface (API) services are used to serve analytics information to different endpoints, thus creating the functional layer of an API gateway which extends orchestration, routing and authorization services.

The orchestration of new processing services as well as API services, is handled by the control service. This service can be accessed via an API and has information regarding available services, as well as already deployed services. It starts and stops services and offers information about health and performance. It is possible to subdivide this control service into smaller less sophisticated services, hence following the microservice paradigm with more rigour.

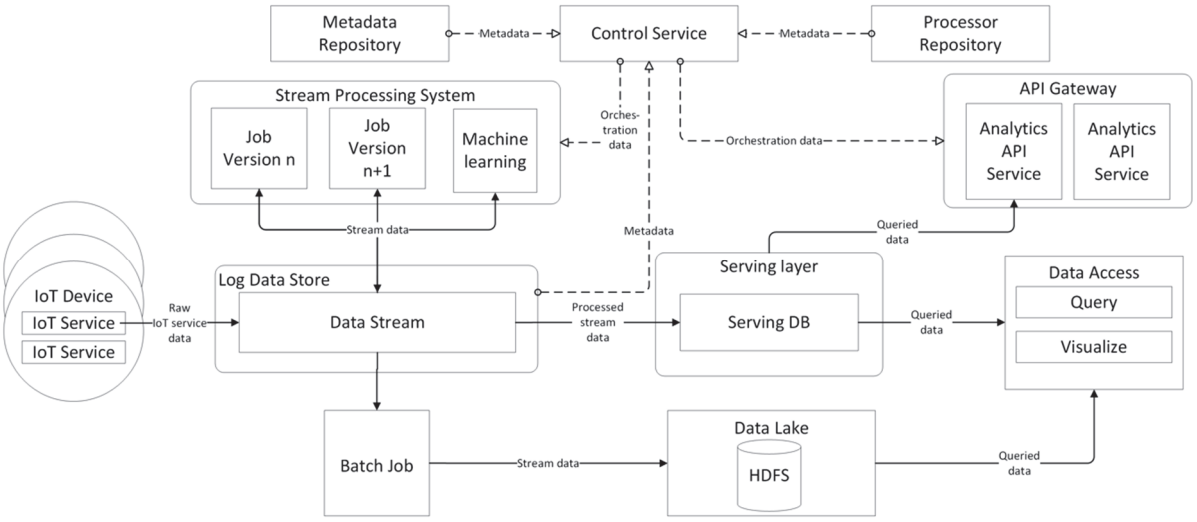

Figure 1: Solution proposal and data flows.

Although, IoT analytics are most powerful when used in a near real-time environment, it is still important to enable users to access historic data.
The log data store as embedded in our approach should only allow data retention of a couple of weeks. Otherwise, reprocessing data in case of changed requirements or needed insights becomes too cumbersome. More specific, it is advised to set retention rates of topics depending on the data ingestion velocity. To overcome data loss, when longer data retention rates are necessary, we use a Data Lake. In order to have the data pushed into it, a batch job implemented as a high-level log data store consumer is triggered at a regular time interval.

As a first step to evaluate the feasibility of our approach, we implemented important parts of the proposed platform architecture. The log data store is provided using Apache Kafka in congestion with Kafka Streams as stream processing library embedded in microservices written in Java. The serving layer consists of the column-oriented data store Druid. Analytics API services are written in Python also designed as microservices. Data visualization is achieved using Metabase.

6 CONCLUSIONS AND OUTLOOK

In this paper, we presented a solution architecture for IoT analytics in the context of personal analytics. This architecture is based on the concept of the Kappa Architecture and uses microservices to enable flexible lightweight stream processing as well as analytics capabilities. Important parts of this architecture have already been implemented but lack automatic orchestration and creation of analytics pipelines. We showed that current IoT analytics architectures are not as well suited for huge numbers of inherently different analytics jobs which change frequently in requirements and semantics. The proposed architecture was designed to overcome these shortcomings. With the future implementation of a Data Lake and the corresponding tools and technologies, we are confident to also provide analytics capabilities which enhance the current ones to be able to handle Big Data problems in terms of volume and velocity as well as variety. Also, the Kappa Architecture itself, by being able to scale processing jobs horizontally, should be beneficial for Big Data real-time processing but needs to be evaluated in this regard.

Further research in this field and more specific on this new type of analytics architecture needs to focus on how to automate data processing further in terms of deployment of processing jobs and the alignment of their inputs with IoT data structures. The use of semantic technologies seems promising to do so and some research has already been conducted (Qanbari et al., 2015). Also, the efficient incorporation of historic data analytics is a key aspect of future research. Therefore, in our next research steps, we will further design and develop the control service as well as the Data Lake to achieve automatic orchestration of data analytics pipeline components on the one side and historic data persistence and insights on the other. Moreover, innovative user interfaces need to be developed to empower consumers to map their own analytics scenarios to the analytics architecture.

ACKNOWLEDGEMENTS

The work presented in this paper is partly funded by the European Regional Development Fund (ERDF) and the Free State of Saxony (Sächsische Aufbaubank - SAB).

REFERENCES

Accenture. (2016). Igniting Growth in Consumer Technology. Retrieved from https://www.accenture.com/_acnmedia/PDF-3/Accenture-Igniting-Growth-in-Consumer-Technology.pdf.

Alrehamy, H., & Walker, C. (2015). Personal Data Lake With Data Gravity Pull. In *2015 IEEE Fifth International Conference on Big Data and Cloud Computing (BDCloud)* (pp. 160–167).

Chang, H.-T., Mishra, N., & Lin, C.-C. (2015). IoT Big-Data Centred Knowledge Granule Analytic and Cluster Framework for BI Applications: A Case Base Analysis. *PloS one, 10*(11).

Cheng, B., Longo, S., Cirillo, F., Bauer, M., & Kovacs, E. (2015). Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander. In B. Carminati (Ed.), *2015 IEEE International Congress on Big Data (BigData Congress). New York, USA* (pp. 592–599). Piscataway, NJ: IEEE.

Choe, E. K., Lee, N. B., Lee, B., Pratt, W., & Kientz, J. A. (2014). Understanding quantified-selfers' practices in collecting and exploring personal data. In M. Jones, P. Palanque, A. Schmidt, & T. Grossman (Eds.), *The 32nd Annual ACM Conference on Human Factors in Computing Systems* (pp. 1143–1152).

Fang, H. (2015). Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER),* (pp. 820–824).

Fetzer, C. (2016). Building Critical Applications Using Microservices. *IEEE Security & Privacy*, 14(6), 86–89.

Greenough, J. (2016). How the 'Internet of Things' will impact consumers, businesses, and governments in 2016 and beyond. Retrieved from http://www.businessinsider.com/how-the-internet-of-things-market-will-grow-2014-10?IR=T.

Hasan, T., Kikiras, P., Leonardi, A., Ziekow, H., & Daubert, J. (2015). Cloud-based IoT Analytics for the Smart Grid: Experiences from a 3-year Pilot. In D. G. Michelson, A. L. Garcia, W.-B. Zhang, J. Cappos, & M. E. Darieby (Eds.), *Proceedings of the 10th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities*.

Jaramillo, D., Nguyen, D. V., & Smart, R. (2016). Leveraging microservices architecture by using Docker technology. In *SoutheastCon 2016.* (pp. 1–5).

Kreps, J. (2014). Questioning the Lambda Architecture. Retrieved from https://www.oreilly.com/ideas/questioning-the-lambda-architecture.

Lewis, J., & Fowler, M. (2014). Microservices: a definition of this new architectural term. Retrieved from http://www.martinfowler.com/articles/microservices.html.

Mineraud, J., Mazhelis, O., Su, X., & Tarkoma, S. (2015). A gap analysis of Internet-of-Things platforms. arXiv preprint arXiv:1502.01181.

Mishra, N., Chang, H.-T., & Lin, C.-C. (2015). An IoT Knowledge Reengineering Framework for Semantic Knowledge Analytics for BI-Services. *Mathematical Problems in Engineering, 2015*(1), 1–12.

Naqishbandi, T., Sheriff, I. C., & Sama, Q. (2015). Big Data, CEP and IoT: Redefining Holistic Healthcare Information Systems and Analytics. *International Journal of Engineering Research & Technology, 4*(1).

Qanbari, S., Behinaein, N., Rahimzadeh, R., & Dustdar, S. (2015). Gatica: Linked Sensed Data Enrichment and Analytics Middleware for IoT Gateways. In *2015 3rd International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 38–43).

Pasupuleti, P., & Purra, B. S. (2015). Data Lake Development with Big Data: Packt Publishing.

Ramakrishnan, R., & Gaur, L. (2016). Smart electricity distribution in residential areas: Internet of Things (IoT) based advanced metering infrastructure and cloud analytics. In *2016 International Conference on Internet of Things and Applications (IOTA)* (pp. 46–51).

Riggins, F. J., & Wamba, S. F. (2015). Research Directions on the Adoption, Usage, and Impact of the Internet of Things through the Use of Big Data Analytics. In *2015 48th Hawaii International Conference on System Sciences (HICSS)* (pp. 1531–1540).

Rozik, A. S., Tolba, A. S., & El-Dosuky, M. A. (2016). Design and Implementation of the Sense Egypt Platform for Real-Time Analysis of IoT Data Streams. *Advances in Internet of Things, 06(04),* 65–91.

Ruckenstein, M. (2014). Visualized and Interacted Life: Personal Analytics and Engagements with Data Doubles. *Societies, 4*(1), 68–84.

Stolpe, M. (2016). The Internet of Things: Opportunities and Challenges for Distributed Data Analysis. *ACM SIGKDD Explorations Newsletter*, 18(1), 15–34.

Swan, M. (2012). Sensor Mania!: The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0. *Journal of Sensor and Actuator Networks, 1*(3), 217–253.

Ueda, T., Nakaike, T., & Ohara, M. (2016). Workload characterization for microservices. In *2016 IEEE International Symposium on Workload Characterization (IISWC)* (pp. 1–10). IEEE.

Wingerath, W., Gessert, F., Friedrich, S., & Ritter, N. (2016). Real-time stream processing for Big Data. *it - Information Technology*, 58(4).

Xu, Q., Aung, K. M. M., Zhu, Y., & Yong, K. L. (2016). Building a large-scale object-based active storage platform for data analytics in the internet of things. *The Journal of Supercomputing, 72*(7), 2796–2814.