Vibration Measurement and Visualization in Semiconductor AMHS on the basis of IoT

Thomas Wagner
Institute of Applied Computer Science
Technische Universität Dresden
Dresden, Germany
thomas.wagner2@tu-dresden.de

Jonathan Seitz
Conimon GmbH
Dresden, Germany
jonathan.seitz@conimon.de

Germar Schneider
Infineon GmbH & Co. KG
Dresden, Germany
germar.schneider@infineon.com

***Abstract*—In this paper, we present an approach to automate a legacy measurement device used for offline vibration measurement within automated material handling systems (AMHS) of semiconductor manufacturing plants by using a modern, state of the art IoT framework. After outlining the drawbacks of the existing, time-consuming procedure of offline measurement, the decision of automating the device using the IoT is explained and the necessary steps and framework services are introduced. Finally, the results and benefits of using an IoT framework as well as the new, automated workflow are documented.**

Keywords—vibration analysis, factory automation, IoT, Arrowhead Framework, AMHS

I. Motivation

Over the last years, the usage of semiconductor products has grown significantly, especially regarding applications that increasingly often appear in our daily lives, like smart phones, intelligent electronic devices, technologies for home automation, automotive devices and renewable energies. One of these broadly used applications are power semiconductor devices. In recent years, Infineon was the first supplier, which started the large-scale manufacturing of these devices in Europe with a successful transfer to a 300mm production facility ensuring manufacturing effectiveness and competitiveness.

However, the change to larger wafer diameters also has increased the risk of production losses by, e.g., wafer cracks due to vibrations, since power devices must be produced on thin wafers which are very sensitive in handling [1]. Therefore, a measurement device was developed that allows the precise monitoring of possible causes of vibration and shocks, e.g., within heterogeneous, automated material handling systems (AMHS) of the wafer fabs. However, as outlined in the next chapter, the process of measurement is time consuming and inconvenient, which is mostly based on the fact that it involves a lot of manual steps and misses a proper integration into the fabs IT infrastructure. To this date, this prevented the usage of the measurement device on a regular basis to achieve a satisfying coverage of vibration measurements within the large, mostly unsupervised AMHS.

Therefore, the goal of this work was the automation of the measurement process and the integration of the developed solution into existing data visualization software of the industrial partner. This measure is absolutely necessary to be able to regularly and easily perform vibration measurements and to quickly localize the areas of potentially critical vibrations and enact countermeasures. Additional, important criteria also included a secured data transfer, the possibility to easily extend the solution with additional measurement devices as well as other sensors and measured variables as well as an easy access to the software and, related to this, a long term maintainability.

The remainder of this paper is structured as follows. First, in Section 2 a general description of vibration measurement in semiconductor AMHS is given and the measurement device is explained in detail. In Section 3, the current measurement procedure and its drawbacks are described. Possible automation solutions based on IoT frameworks and the appropriate requirements are discussed in Section 4. Section 5 deals with the implementation of the selected solution and highlights some noteworthy aspects. The results of first tests of the implemented solution are evaluated in Section 6. Finally, in Section 7 conclusions are drawn and an outlook is given.

II. Vibration Measurnments in Semiconductor AMHS

The fully automated material handling system is the backbone of modern 300mm semiconductor manufacturing plants, since it ensures that lots are being quickly and effectively transported to the correct production equipment. A number of variations for transport systems are available today, ranging from overhead hoist transportation systems, rail guided vehicles, automated guided vehicles to conveyor-based systems.

Fig. 1: Overhead Hoist Transport (OHT) Vehicle

The issue of critical vibrations occurring during transports can more or less be expected for all of these systems, but is especially crucial for hybrid systems, where several of these systems are interconnected. These connections are especially prone to shocks if they are, e.g., not sufficiently aligned. Such hybrid systems are necessary where architectural circumstances, e.g., different ceiling heights, do not allow for the continuous usage of a single system.

Especially for 300mm thin wafers and the substrates required for their production, it is important that the transport is as vibration-free as possible [2]. Critical are vibrations that occur in the range of the natural frequency of wafers or FOUP and thus can lead to resonances as well as generally shocks above 2g [3]. Shocks exceeding 4g can cause considerable damage to the wafers and are classified as very critical.

In order to measure and compare such vibrations during wafer transport, different measuring methods are available. In stationary measurement, shocks are applied to the FOUP using a shaker as vibration exciter and the resulting vibration of the FOUP container is measured by means of a laser vibrometer. Thus, as described in [3], the natural frequency of the FOUP can be determined and appropriate damping measures can be taken. In this way, the breaking point of the wafers in a FOUP filled with wafers can also be determined [4] [5]. These values can be used as a reference for the dimensioning of the limit values in the following measurement procedures. However, this measuring method is not suitable for permanent monitoring of the transport system, since measurements show that the actual vibrations in the transport system due to wear and incorrect calibration cannot be predicted precisely and change over time. To record the actual vibrations during the transport process, the use of external sensors is conceivable. Stationary laser sensors, which are mounted laterally next to the transport rail, can detect the vibrations of a passing FOUP in the transport vehicle and determine the vibration intensity with the help of the laser speckle method [6]. However, this type of measurement only allows a fixed vibration measurement at certain points and is therefore not suitable for the investigation of all transport processes.

Fig. 2: Conveyor based transport system of a 200mm production line

Common to all of the systems described above is the usage of FOUPs (front opening unified pods [2]) for the transportation of wafers. Therefore, a measurement FOUP (ref. Fig. 1) was already developed in cooperation between Technische Universität Dresden (TUD) and the Infineon GmbH & Co. KG (IFD), which is able to monitor critical shocks and vibrations during wafer transportation with the help of a motion sensor and an embedded controller that records and saves information about acceleration, gyroscopic rotation, magnetic field values, rough heading, temperatures as well as absolute time stamps in configurable formats, periods and aggregates. The recorded data is stored with a respective time stamp in CSV format on the internal device memory or on a connected USB flash memory.

This first demonstrator was used to check selected components of the 300mm AMHS and to compare the results to manual wafer handling. The vibration measurement with the Measurement-FOUP is the basis of the measures implemented within the scope of this work. The direct and mobile data acquisition of the Measurement-FOUP is best suited to monitor automated transport systems across the board. However, the operation of the device is quite cumbersome and the data evaluation is error-prone and time-consuming due to many manual steps. The following chapter describes in detail the sequence of a measuring run, the shortcomings in the operation of the Measurement-FOUP and the data transmission.

Fig. 3: Measurement device built into a FOUP

III. Existing Measurnment Procedure

The main issues in operating the Measurement-FOUP originated from the fact, that the recorded data had to be downloaded and processed manually, therefore, the data post processing, e.g., identification of critical vibrations as well as the mapping of vibration measurements to actual locations within the AMHS, was difficult and only feasible for isolated, small test measurements.

In detail, the following steps must be taken for a successful measurement run and evaluation. First, the measuring FOUPs is switched on, the measurement now starts immediately depending on the configuration or the start of the measurement must be initiated by pressing a button. From then on the vibration data is recorded and written into a CSV file. While the measurement is running, the FOUP is transferred to the AMHS via a stocker (intermediate storage for FOUPs as part of the transport system). Via the user interface of the AMHS, the destination of the measurement run can now be entered. The FOUP is picked up by an OHT vehicle and driven to the corresponding destination. Depending on the destination, it may be necessary to change vehicles during the trip. If, for example, it is located in a part of a building that can only be reached via a conveyor (see Fig. 6), the measuring FOUP is placed on the conveyor by the first vehicle and then picked up and transported by another vehicle. The operator now has to walk to the destination stocker and instruct there again via the stocker software that the FOUP is transported to the output port. Then it can be removed and the measurement can be finished.

For further processing the data must be copied manually to a computer. The evaluation was carried out by using a proprietary visualization software demonstrator that was developed in the context of a previous work (cf. Fig. 4). With this software the data can be imported and are displayed in a common diagram, separated by direction of impact. Furthermore, the route of a measurement run can be displayed.

For this purpose, the log files of the transport system have to be manually imported from the corresponding servers are loaded and imported. The program filters these according to the vehicle data entered and displays them in a coordinate system. System experts can see from the grid of points which route the measuring FOUP has taken.

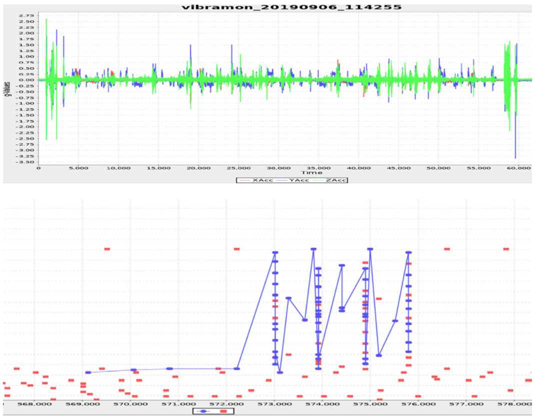

Fig. 4: Proprietary visualization tool

Only a rough area delimitation is possible with this method. However, the user must have detailed knowledge of the transport system and has to make many assumptions. Experience has shown that if a measurement run exceeds a duration of approx. five minutes, it is almost impossible to perform a useful delimitation.

To alleviate these issues, both the measurement runs and the collection, post processing and evaluation of the data had to be automated as much as possible.

IV. The Use of IoT Frameworks in the Industrial Context

Decisive for an effective use of the Mess-FOUP is an easy handling, an automated data transfer and processing as well as the informative visual presentation of the data. In order to be able to further process the sensor data, such as those collected by the measurement FOUP, they must be transferred from the data source to the data sink. In the present application, the microcontroller of the FOUP is the data source and a map-based visualization program within the IT landscape of the semiconductor production is the data sink. The data can be transferred manually, as described in the previous chapter. However, this procedure has many disadvantages and can be automated, standardized and accelerated with the help of an IoT (Internet of Things) framework, or, more specifically IIoT (Industrial Internet of Things) in the industrial setting.

Such software frameworks provide general solutions for recurring tasks. For this purpose, the problem is generalized and abstracted in order to apply the solution as generically as possible to many different but similar problems. The interfaces between framework and concrete application are clearly defined. The framework itself is not yet a finished program, but provides a template with which similar problems can be solved in a uniform and reusable way. Almost all IoT applications require communication between data sources and the processing components. Therefore, the use of special IoT frameworks that take over this task makes sense. IoT frameworks almost always are used to model a system of heterogeneous subsystems, a so-called "System of Systems (SoS)". According to Maier [7], System of Systems differ from a large, monolithic software system [8] in the following aspects, that are all relevant for the application at hand:

Operational independence of the elements: The measurement FOUP, other data sources and the visualization software can act independently of each other.

Management independence of the elements: Even after the integration of the elements into the overlying system, functions remain, which are only executed within the individual component.

Evolutionary development: The SoS has no final state. Components are added, removed or changed depending on new or changing requirements. Even if this is not yet the case in the current use-case, further data sources for later integration into the present use-case are conceivable.

New functions: By linking the elements to form a new system, new functions or information are created that cannot be generated by a single element of the system.

Physical separation: It must be possible to create a physical separation of the elements without losing functionality of the SoS, since the components are operated on physically separate devices, use different operating system or are located in different network environments.

There is a wide variety of available IoT and IIoT frameworks, e.g., AUTOSAR, IoTivity, BaSyx, the Arrowhead Framework, FIWARE or LWM2M, most of which are tailored to specific applications. With regard to the selection of a suitable framework, there are a number of requirements to consider. For example, continuous new developments are responsible for the fact that new data sources (e.g. new sensors) and -sinks (e.g. new monitoring tools) have be integrated into the IoT SoS in future. In addition, the functionality of existing components will be extended by further services, some of which require additional data sources, e.g. for aggregating different sensor data. Many of these use cases cannot yet be concretely foreseen when the framework goes live. Nevertheless, it should be as easy as possible to integrate or replace components afterwards. The specific adjustments on the side of the framework should be kept as low as possible. Therefore, a generic interface definition is necessary. This is the only way to achieve interchangeability of individual components.

Since IoT is integrated into an already existing IT infrastructure, the migration and adaptation effort must be minimal. To ensure error-free data transmission, it is important to comply with standards. This refers both to the underlying technical transmission protocols and to the type of data preparation. Often, established standards and transmission formats are used and a data transmission scheme is defined based thereon. The most frequently used transmission formats are XML and JSON. If additional sensors, data sources or data sinks are integrated into the framework, the IoT framework must also be able to grow. Here, the framework must offer possibilities to group and divide up the existing functionality and the underlying infrastructure. If several registered services are involved, the framework must still be able to respond reliably and quickly.

An SoS based on IoT Frameworks is made up of heterogeneous subsystems. In order to be able to manage them at runtime, it must be possible to include or remove subsystems at runtime. For this purpose, couplings between services must be created and resolved dynamically (loose coupling). Services that connect to each other at runtime might not know each other at system startup (late binding). For instance, the presented use-case is based on the measurement data of the measurement FOUP. This device is not permanently switched on due to the power supply via a battery. Runtime capability is therefore essential to be able to integrate and to couple it with other services as soon as it is switched on. In the same way it must be possible to switch off the device as a subsystem and to stop its offered services without malfunctions. Runtime capability is a requirement that many IoT frameworks do not or only insufficiently fulfill [9], since this is not necessary in many application areas.

An increasingly important criterion is the security of the framework. It must be ensured that only authorized services can be registered and only appropriately authorized services can use the data. For this purpose, there must be an authentication authority, that authorizes or denies services and consumers. A connection between the services may only be established after both sides have been verified. In addition, the subsequent communication between the services must also be implemented in a tamper-proof manner. In the concrete application case, the data exchanged include highly sensitive production data such as product routes, from which important information such as tool sequences, the factory layout and other data worthy of protection could be determined. Therefore, the security of the IoT Framework is of particular importance in this case. On the other hand, in smart home IoT applications, this requirement is given much lower priority. Here, the data is usually transmitted over the Internet, sometimes without encryption [10].

In addition to the need for platform independence, already mentioned as "physical separation", the aspects of the framework development stage and support are highly important for use in productive environments. Especially due to the high demand in countless new application areas, new IoT Frameworks are increasingly being developed. However, many of these frameworks often do not meet the requirements of the industry, who usually prefer a stable, mature and proven software to a possibly technically superior but not established framework. A large user base means that such a framework will be further developed and maintained. Designs that involve large companies from the start of the project, who want to use these frameworks themselves, therefore have a much better chance of establishing themselves in the market.

Another advantage in this context is the publicly availability of the source code (open source). In open source projects, users are able to make individual changes to the software themselves. With closed source frameworks, the customer becomes more dependent on the software vendor, because he has to trust the vendor of the software that the framework will be further developed. Furthermore, the customer can only check to a limited extent whether, for example, all security requirements have been implemented correctly. Related to this, an often somewhat neglected requirement is the documentation of the IoT framework. To make it easier for third parties to use the framework, it must be documented in a structured way. Poorly documented frameworks can only be maintained by experts in the respective system and have almost no chance of becoming established in the long term.

Considering all these requirements, the authors decided to rely on the Arrowhead Framework for the implementation of the use-case. Apart from some remaining shortcomings in the current documentation, the Arrowhead Framework realizes the described requirements very well (also refer to [9] for a comprehensive comparison).

V. AUTOMATION OF THE MEASUREMENT PROCEDURE

Arrowhead is an IIoT framework that enables the automated exchange of information between any IoT components. For this purpose, the functions of the systems are abstracted into services and bundled in so-called local clouds [11]. The Arrowhead framework consists of the core systems and the application systems. The former are an integral part of Arrowhead. A distinction is made between necessary core systems (mandatory core systems) and optional core systems (automation support core systems). The applicability of the optional core systems depends on the respective use case. The application systems are use case specific and must therefore be implemented by the user. For this purpose, the core systems offer predefined communication interfaces. If several local clouds are linked together, each local cloud has its own core and application systems.

The use case at hand has been implemented as a local cloud within the Arrowhead framework according to the schematic depicted in Fig. 5. As stated above, the mandatory core system are the basic services of this cloud. The Service Registry provides a directory of all available services within the respective local cloud. Services can be added, changed or deleted with the help of corresponding requests. The registered services can be found by potential consumers via the Service Discovery Service (DNS-SD). In addition, the Service Registry can also contain priorities in the form of service weightings. This information can later be taken into account by the orchestrator. The Authorization System offers the services "Authorization Management" and "Authorization Control" and ensures that only authorized consumers may use a service. For this purpose, authorizations for the services stored in the service registry can be created. Precise access rules for the services can be defined via the Authorization Management Service.

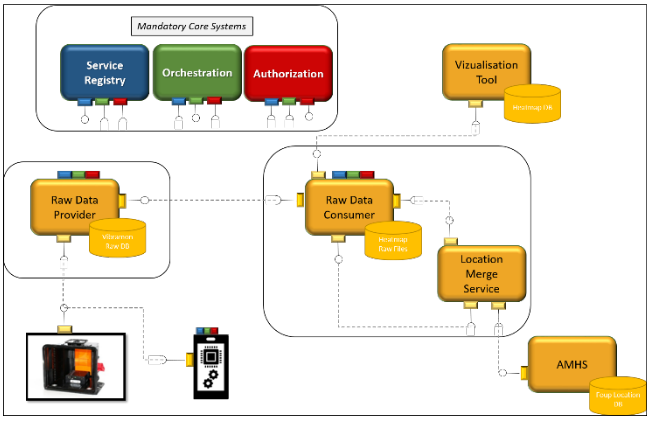

Fig. 5: Schematic of the vibration measurement components within the Arrowhead framework

The Orchestration System is the central element for the reuse of services. For this purpose, requirements (QoS) can be defined manually or via a coupled planning tool and transferred to the Orchestrator. The requirements are compared with the service information of the service registry and appropriate service bindings are created. These bindings can be created and removed dynamically using the principles of "loose coupling" and "late binding". Thus, changes can be implemented at system runtime without the need to restart the system.

In addition to the mandatory core systems, some of the many available “automation support core systems” have been selected that are relevant to the use case (or logical extensions). For example, the QoS Manager system is used to configure and monitor Quality of Service requirements. It extends the functionality of the Orchestrator system by the possibility to define quality rules, which are necessary for the coupling of services. These can be minimum sampling frequencies, for example. The Orchestrator system then only creates service couplings where the required restrictions are implemented. With the Event Handler System, an event-driven process can be implemented. For this purpose, it offers a list in which subscribers, i.e. services that listen to certain events, can be entered. As soon as a corresponding event is created by the Event Publisher, it is forwarded to the Subscriber Services, which can then react to it. Lastly, the Workflow Choreographer System offers the possibility to define a time schedule of services ([12], [13]). Service sequences, called recipes, can be defined for this purpose. These are implemented using the orchestrator and event handler system. The information about the progress of the service is stored directly within the events. Once a service is finished with its processing part and sends a corresponding event, the following service in the recipe can be executed.

In the following, the services "raw data provider (RDP)", "raw data consumer (RDC)" and "location merge service (LMS)", which are colored yellow in Fig. 5, are described. The component "Visualization" is an externally connected, map based visualization tool in the fab, "AMHS" represents the databases of the "Overhead Hoist Transport Controller (OHTC)", which orchestrates the transport orders and vehicles in the AMHS and logs a lot of information about the vehicles, e.g. their position, in real time.

The files created by the measurement FOUP are stored locally on the device memory after a measurement run. However, since it is not permanently switched on, it cannot act directly as a service provider in the Arrowhead Framework, instead a permanently available intermediary is required, the RDP. It checks cyclically if the measurement FOUP is online and if there is new data present. If so, the data is copied into the storage of the RDP. When the raw data is imported, the CSV files are parsed into Data Transfer Objects (DTO). Since the RDP is registered in the Service Registry as a publisher service and the RDC is authorized as a subscriber to the RDP's services, it is automatically notified to the presence of new data. The RDC parses this data and, besides from logging axis-related vibration values (Fig. 6), calculates a gravity-adjusted force vector $\mathbf{R}_{\mathbf{res}}$ according to the following formula:

$$\boldsymbol{R_{res}} = \sqrt{(Ax)^2 + (Ay)^2 + (Az+1)^2}$$

This vector is needed to display the vibration data in the visualization tool as compact as possible. By calculating $\mathbf{R}_{\mathbf{res}}$, even impacts that do not follow the movement axes exactly are displayed correctly. Following this calculation, the RDC starts the LMS, which uses the respective time stamps to merge the vibration measurement values of the measurement FOUP and the location data of the OHTC. The location data stored in the OHTC consists of coordinates in a proprietary coordinate system that is used by many applications within the factory.

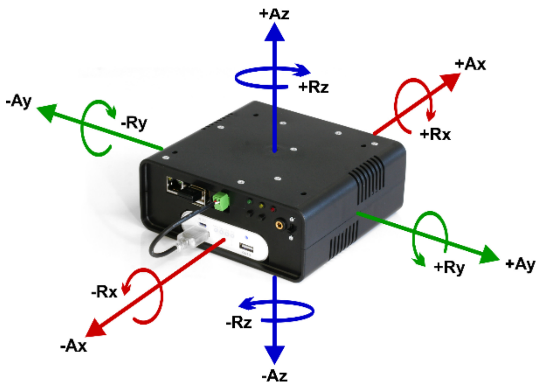

Fig. 6:Directions of the acceleration (A) and rotation axes (R) in alignment with the measurement FOUP

In order to find the data of suitable trips and vehicles in which the measurement FOUP was transported, the OHTC database is filtered according to a unique identification number of the measurement FOUP. Since the location data is appended based on matching time stamps, it is particularly important that the internal system clock of the measurement FOUP and the clock of the OHTC system run synchronously.

The merged files are cached by the RDC and passed on to the visualization tool if required. There they are read automatically and presented to the user on request, as described in the following chapter.

VI. Results and Evaluation

By using the Arrowhead IoT Framework, automated vibration monitoring of the entire transport system is possible for the first time. The implementation saves significant time and money, while providing the interface to reuse both hardware and software components. Users benefit from the reduced complexity of the process and the informative and easy-to-understand visualizations. Due to the automation of the process, the time required to perform a measurement run was reduced from 2h to 20 to 30 minutes (see Fig. 7). This represents an improvement of 83%. In addition to this significant economic factor, the number of people who can carry out a measurement run from a technical, professional and physical point of view has also been increased. Previously, only an engineer with expertise in computer science, data processing and vibration analysis could carry out a measurement run.

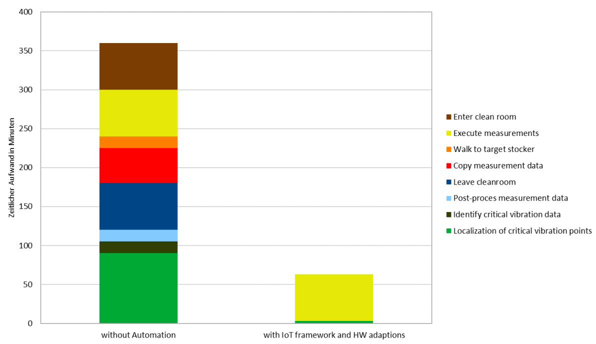

Fig. 7: Comparison of the time requirements before and after the automation

Fig. 8 shows an example of the resulting heat map of a measurement run as it is shown in the visualization tool. Clearly visible is the significant improvement of the representation as well as the increased information content compared to the previous, proprietary solution (Fig. 4). The solid colored line shows the route taken in the factory. The color of the line indicates the maximum vibration level that has occurred in this section of the route. This makes it possible to see at first glance where a source of problems causing critical vibration values is located. The user can select individual vibration axes for the coloring of the line by further configuration. The expert thus receives precise information about the location, strength and type of a fault and can immediately take appropriate countermeasures. The green lines mark sections where the vibrations were below the threshold values. The red lines show sections where critical vibration values have been reached and corresponding measures must be taken. The information from the generated heat maps can be used to take targeted preventive maintenance measures that make modern semiconductor production safer, more ecological and more economical.

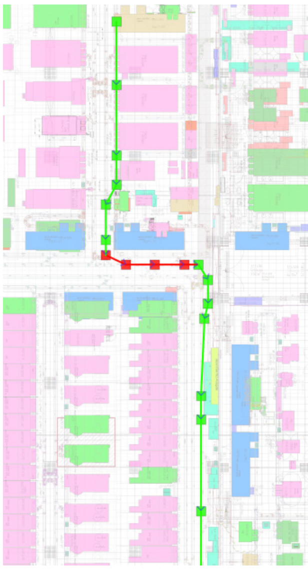

Fig. 8: Heat map of a measurement run of the with aggregated vibration data

The developed solution can also be applied to comparable automated transport systems with vibration-sensitive goods or materials. For example, automated transport systems in parcel centers, airports or picking centers could also benefit from an automated vibration detection and visualization.

VII. Conclusion and Outlook

The automation of the measurement FOUP achieved using the Arrowhead IoT framework has undoubtedly made work easier and saved engineering efforts, time and money. The strengths and advantages of the framework compared to a proprietary software solution adapted to the specific application are particularly evident when a large number of services are integrated. Arrowhead standardizes the coupling and orchestration of the services and thus simplifies subsequent changes. A reuse of already developed components is easy. However, if the measurement FOUP data is to be used only for the original task, i.e., only to be displayed with the visualization tool, a solution without IoT Framework would be easier to implement. In this case, framework specific services can be omitted, since it is a one to one assignment of data source and data sink.

With fewer intermediate stations, data transfer would be faster and less error-prone. In the case of measurement FOUP, the use of the Arrowhead Framework is only reasonable if additional sensors or services are integrated into the IoT framework. Only the use of several providers and consumers in the existing framework outweighs the additional effort required to install the Arrowhead Framework.

Therefore, it will be evaluated in future if other services could be integrated into the local cloud, for instance, sensor to monitor the machine health of production- or facility equipment.

Acknowledgment

The co-funded innovation project Arrowhead-Tools receives grants from the European Commissions H2020 research and innovation programme, ECSEL Joint Undertaking (project no. 826452), the free state of Saxony, the German Federal Ministry of Education and national funding authorities from involved countries

References

[1] G. Schneider, T. Wagner, and M. Kraft, "Use of Simulation Studies to Overcome Key Challenges in the Fab Automation of a 300 mm Power Semiconductor Pilot Line Comprising Thin-Wafer Processing", 26th SEMI Advanced Semiconductor Manufacturing Conference (ASMC 2015), Saratoga Springs USA, 2015.

[2] W. L. Tsai, H. H. Chang, C. H. Chien, J. H. Lau, H. C. Fu, C. W. Chiang, T. Y. Kuo, Y. H. Chen, R. Lo, and M. J. Kao. How to select adhesive materials for temporary bonding and de-bonding of 200mm and 300mm thin-wafer handling for 3D IC integration? In 61st IEEE Electronic Components and Technology Conference (ECTC), 989-998, 2011.

[3] T. Müller, T. Schmidt, S. Rank, and G. Schneider. Natural frequencies determination of wafers. In Logistics Journal: Proceedings 2017(10), 2017.

[4] J. Steele and T. Biswas. Development of a shock & vibration spec for 300mm wafer AMHS handling. In 17th Annual SEMI/IEEE ASMC 2006 Conference, pages 245-250. IEEE, 2006.

[5] J. Stoutenburg and R. Howell. 300 mm One-Way Front Opening Shipping Box. In Proceedings of the International Symposium on Semiconductor Manufacturing, pages 443-446, 2003.

[6] B. Ruth. Speed and vibration measurements with the laser speckle method (in german: Geschwindigkeits-und Vibrationsmessungen mit der Laser-Speckle-Methode). In Laser/Optoelectronics in Engineering (in german: Laser/Optoelektronik in der Technik), pages 177-180, Springer, 1987.

[7] M. W. Maier. Architecting principles for systems-of-systems. Systems Engineering, In Journal of the International Council on Systems Engineering, 1(4): 267-284, 1998.

[8] Delsing, J. (Ed.). IoT Automation. Boca Raton: CRC Press, https://doi.org/10.1201/9781315367897, 2017.

[9] C. Paniagua and J. Delsing. Industrial Frameworks for Internet of Things: A Survey, In IEEE Systems Journal, 2020.

[10] D. Bastos, M. Shackleton, and F. El-Mousa, Internet of Things: A Survey of Technologies and Security Risks in Smart Home and City Environments, In Living in the Internet of Things: Cybersecurity of the IoT, London, 2018.

[11] J. Halme et. al., Monitoring of Production Processes and the Condition of the Production Equipment through the Internet, In 6th IEEE International Conference on Control, Decision and Information Technologies - CoDIT 2019, Paris, 2019.

[12] I. Marcu, G. Suciu, C. Bălăceanu, A. Vulpe, and A.-M. Drăgulinescu, Arrowhead Technology for Digitalization and Automation Solution: Smart Cities and Smart Agriculture. In Sensors, 20(5):1464, 2020.

[13] P. Varga et. al., Making system of systems interoperable-The core components of the arrowhead framework. In Journal of Network and Computer Applications, 81:85-95, 2017.