percepio®

Improving the Reliability of the Internet of Things

Navigate the complex challenge of anomaly detection and reporting, cloud management and over-the-air updates, for effective problem solving in the field.

percepio®

Improving the reliability of the Internet of Things

Systematic anomaly detection in firmware enables improved dependability during operation of IoT device fleets. Dr. Johan Kraft, CEO and founder of Percepio, explains the benefits.

Despite the best efforts of developers, embedded and IoT systems are usually deployed with bugs remaining in their code. A development team introduces an average of 120 bugs per 1,000 lines of code during development. Approximately 5 percent, or 6 bugs per 1,000 lines of code, typically remain in the shipped software.

When there are thousands of IoT devices deployed in the field, relying on users to report problems caused by these bugs is neither reliable nor scalable. User reports tend to be vague and unhelpful for problem solving. When there are millions of devices, this matters even more.

A system for detecting and reporting bugs and anomalies should be a key part of your toolbox. Coupled with software tracing* and the ability to do quick over-the-air (OTA) updates, this becomes a powerful mechanism to improve the quality and reliability of a wide range of embedded and IoT systems.

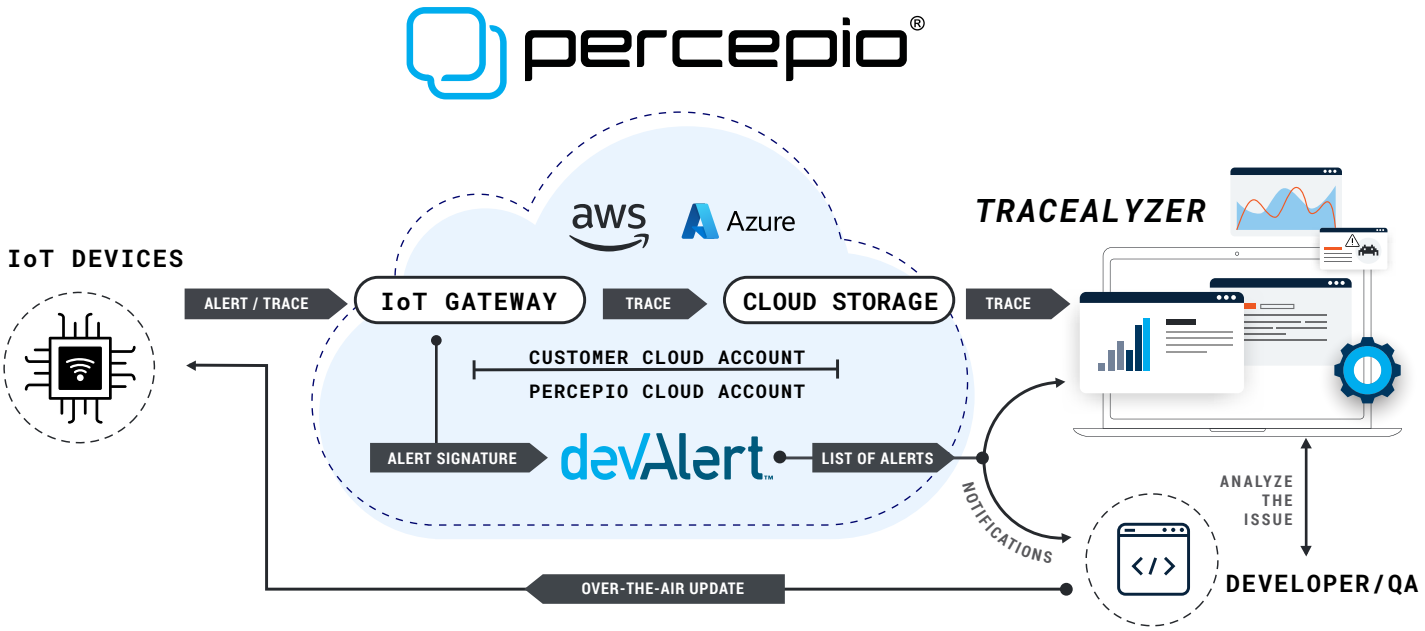
* Software tracing refers to systematic recording of timestamped software events from firmware in an embedded or IoT device. Events can be recorded from kernel code, user application code, or both. Used together with a visual trace analyzer tool, such as Percepio Tracealyzer, software tracing provides unsurpassed insight into the dynamic behavior of an embedded system.

These lingering bugs probably won't show up immediately, but only cause problems under certain circumstances. Otherwise, they would have been found before the product shipped. While an OTA update can solve the problem in the field, developers and operations staff need a feedback system to identify issues in deployed devices in near real time. This approach has long been standard in the development of mobile and cloud applications (DevOps) and it has now become viable for embedded development as well.

Developers and operations staff need a feedback system to identify issues in deployed devices in near real time.

The key to discovering — and solving — problems in the field is a combination of detection tools, software tracing, cloud management and OTA updates. This is a complex challenge. Tracing code needs to be as efficient as possible in a system already constrained in resources. The link to the cloud needs to be secure and transparent. It also needs to transfer the right data to help developers identify any problems quickly and easily. The cloud service should identify which issues are new and important, and then notify developers of the problem they need to fix. Once it's fixed, the updated software must be distributed to all devices. All of this needs to scale across millions of devices.

Information flow starts in the error handling code of the IoT device, such as existing sanity checks and fault exception handlers. Using a software agent, firmware issues are shared as alerts to a customer's cloud account. An alert may include an error message and any other information relevant to the specific issue, such as software state variables and hardware registers. Depending on the severity of the issue, the alert is either shared directly or after a device restart, once the cloud connection has been restored.



Alerts may also include a trace of the most recent software events in the device, recorded automatically by the agent. The trace provides details of the error and the context, making it easier for developers to identify the bug.

Encoding efficiency is key. The developers need enough context to identify the real problem, but tracing should be done using a minimal amount of memory.

This is important for two reasons:

» It reduces the upload time to a fraction of a second, and

» It minimizes the cloud-side operational costs of alert messaging and storage.

This encoding efficiency makes it possible to use trace technology out in the field, even in small IoT devices, bringing dramatic advantages.

Alerts from the firmware agent are uploaded to the customers' cloud service. This is configured to store alerts and notify the DevAlert monitoring service. DevAlert then handles classification, statistics and notifications to the developers. Classification reduces the potentially large stream of alerts into a managable overview of issues to investigate, creating a to-do list that ensures nothing is missed.

This service offers configuration options, identifying, for example, the conditions under which notifications should be sent and to whom. When developers receive notifications about new issues, they can access alerts and traces to see the problem.

Privacy is also key here. The software trace never needs to leave the customer's cloud account. Only an anonymized signature of the alert is required for cloud processing, which can be provided in an external cloud service.

This information can be made completely transparent, configurable, and meaningless on its own. The communication and storage is provided by the existing capabilities in the developer's IoT platform, using best practices for authentication and encryption.

Need to Trace Your IoT Device?

Testing in the lab just isn't enough. Today's embedded IoT systems are complex. To eradicate all software issues you need real time tracing and alerts to identify bugs in the field as they happen, with automatic notifications to the developers to speed up resolution.

Such a system has to be scalable, secure and transparent to the developers. Once in place, it provides immediate awareness on the very first occurrence of an issue — before many users have been affected, and lets developers take full advantage of OTA updates to rapidly improve their product.

ABOUT THE AUTHOR//

Dr. Johan Kraft is CEO and founder of Percepio AB. Dr. Kraft is the original developer of Percepio Tracealyzer, a tool for visual trace diagnostics that provides insight into runtime systems to accelerate embedded software development. His applied academic research, in collaboration with industry, focused on embedded software timing analysis. Prior to founding Percepio in 2009, he worked in embedded software development at ABB Robotics. Dr. Kraft holds a PhD in computer science.