Operations Wisdom Logging

Tomislav Žitnik, Zoran Bosić
Ericsson Nikola Tesla d.d., Zagreb, Croatia
tomislav.zitnik@ericsson.com, zoran.bosic@ericsson.com

Abstract: Good customer experience is one of the key success factors in today's information technology industry and digital transformation processes. High-class customer experience and more efficient business performance achieved through simplified processes with automated actions are building bricks of digital transformation which enables optimized performance. To that end, understanding and maintaining information through knowledge management process makes it easy to capture, query, find, reuse information, and learn from performed actions or events.

Operations wisdom logging (OWL) toolset developed in Ericsson Nikola Tesla Group aims to improve customer experience by knowledge management process and by transforming it into suitable action flow. OWL toolset supports data collection and transformation of analyzed data from information to knowledge required for making sensible decisions. OWL solution comprises a single access connection point with clientless remote access to any remote node which requires daily maintenance. Passive data logging during remote access is transformed into consolidated information and knowledge ready to integrate with any information technology service management (ITSM) solution with quick search engine empowered by optical character recognition (OCR).

Machine Learning/Artificial Intelligence (ML/AI) concept based on predefined events and user profiles provides the context of OWL toolset solution lifecycle, enabling additional value to customer experience during node or equipment maintenance.

Key words: DIKW, ITSM, CMDB, ML/AI, OCR, correlations, knowledge.

I. INTRODUCTION

Operations wisdom logging (OWL) toolset developed in Ericsson Nikola Tesla Group integrates remote access solutions with information technology service management (ITSM) solutions. Within this process OWL toolset supports a new paradigm in data collection and transformation of analyzed data from information to knowledge required for making sensible decisions.

Until recently most of remote access and ITSM solutions have been employed separately but now OWL toolset enables a great benefit of combining and integrating them. OWL toolset model is based on knowledge management (KM) and supporting automated processes. KM is usually displayed within the data-to-information-to-knowledge-to-wisdom (DIKW) pyramid (Figure 1). Data logging is a basic functionality of every information technology (IT) system. It is a passive activity that implies recording of the observed data by their nature design. In case that no one monitors or analyses the data, they will be logged without any purpose, thus creating a heavy load on IT resources and their extensive utilization.

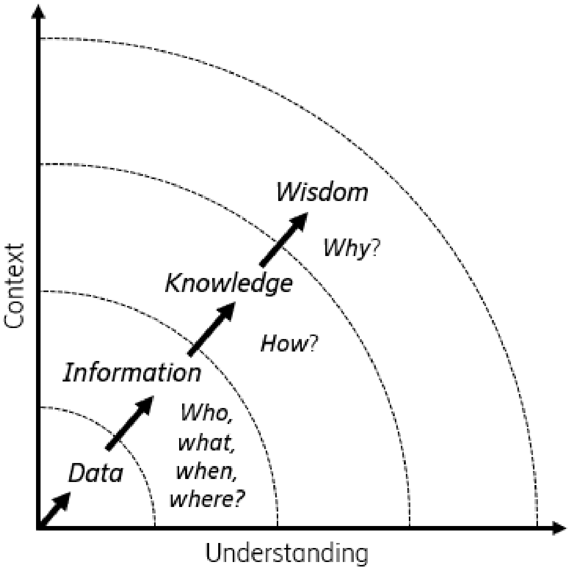

Figure 1. DIKW model - the flow from data to wisdom

OWL toolset focuses on key KM activities around data: collecting accurate data, identifying, and analyzing data, transforming data into valuable information. After data have been transformed into information, the next key step in KM process is managing the information content in a way that makes it easy to capture, query, find, re-use information and to learn from experiences. Knowledge is a context-based process which puts information into easy-to-use form. Wisdom makes use of knowledge to create value through correct and well-informed decisions. Wisdom involves having the application and contextual awareness to provide strong common-sense judgement.

OWL toolset solution design supports each step of DIKW KM process. It offers fully consolidated data in one place, fully automated DIKW process with the complete platform operating independently of the service provider. By integrating with ITSM, the necessary knowledge is captured in one place within the already available ITSM platform, which makes OWL toolset a highly reusable solution based on free open-source software (FOSS) [1]. The use of open-source software in an Ericsson product is defined by the Ericsson Group directive “Free Open-Source Software” [2]. With this approach OWL toolset decreases capital expenditure (CapEx) and operational expenditure (OpeEx), which leads to the positive return of investment (ROI).

II. PROBLEM STATEMENTS

In daily IT business there are multiple teams, like service integrators or operations who maintain IT and telco components, so called end-nodes. End node can be any IT or telco equipment, or an application which requires remote access via application clients or multiple protocols such as SSH, RDP, Telnet, VNC, moshell, mysqul clients, and docker console.

The frequent problem which maintenance teams encounter in daily work with different nodes is slow response if they do not have consolidated and relevant information. For example, when an operations team want to connect to an end node, usually these connection data are scattered in different places. Each individual or team may or may not have consolidated data and this becomes a challenge when the number of end nodes grows on daily basis.

The second problem is occurrence of different jump hosts (JH), terminal servers (TS) like Citrix. Once an operations team member connects to the server, he/she needs to open another remote access application, like putty or moshell to finally connect to the end node. Due to different business needs, like security, such connections are usually logged but these saved logs have limited auditing capabilities. So, the logs exist but the analysis of the data and the correlation between them is mostly absent. This results in high utilization of IT resources, especially disk space, where, in fact, the passively logged data serve no purpose.

The third problem is the ITSM connection itself or the integration of different tools that are necessary to solve a problem. A common example is solving an incident on a node where it is necessary to use monitoring and ticketing tool, knowledge management database (KMDB) and known error database (KEDB) to decide how to solve the problem. At the end, TS or JH are required to connect to the node and solve the problem. This procedure can take a few hours as browsing through the multiple windows and search queries are done manually.

When all these aspects of node maintenance are put together, we end up with reactive and manual support that is quite slow and where improvisation or ad hoc problem solving are common. The slow support response causes dissatisfaction of the end user and it can even lead to non-fulfillment of end node service key performance indicators (KPIs) and service level agreements (SLAs).

III. OWL TOOLSET DESCRIPTION

OWL toolset solution has been developed based on Information Technology Infrastructure Library v3 [4-8] service lifecycle model. ITIL service transition focuses on two things: KM, which includes KMDB and KEDB, and on DIKW pyramid (Figure 1.) In addition to that, ITIL service operations focus on the common service operations activities, such as monitoring and control, event and incident management, and request fulfillment.

OWL toolset solution lifecycle comprises the following key architectural components (Figure 2):

- Remote access solution
- Data collection
- ITSM integration
- AI/ML Control.

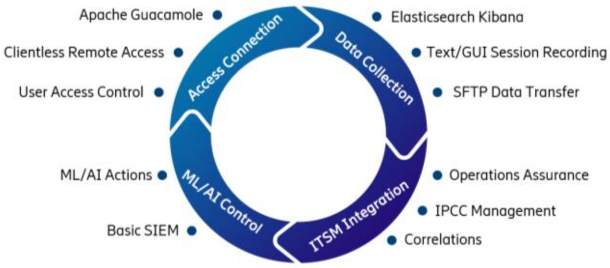

Figure 2. OWL solution lifecycle

A. *Remote Access Solution*

The remote access solution is based on Apache Guacamole solution and represents a central component in OWL toolset. Apache Guacamole is a clientless FOSS solution [3]. A clientless solution means that no client software or plugin are needed to establish the remote connection to an end node. The remote connection is established through the HTML 5 capable browser (Figure 3.). This also means that access can be established from desktop machines, but also from mobile devices such as mobile phones and tablets. This approach offers a possibility to integrate Apache Guacamole with other web applications, and what’s more, an end user does not have to perform any installation or configuration protocols to use it.

Through Apache Guacamole an end user can access its remote nodes using several supported remote protocols like telnet or SSH for command line access or, RDP and VNC for GUI access. Also, the direct access to Kubernetes containers consoles is available. An integrated SFTP client enables easy file upload/download which can be invoked during the session. FTP can be enabled/disabled per session and files can be uploaded/downloaded through the same browser session.

Apache Guacamole access offers several different types of authentications and authorization with the support of single sign on (SSO). Internal users and groups can be used, but it is also possible to integrate external services such as LDAP or active directory. In addition, Apache Guacamole allows a combination of both internal and external authentication. For implementation in Ericsson Nikola Tesla, we have decided to integrate authentication with the corporate active directory. This gives the benefit of employing the standard corporate authentication, as well as controlling access to remote machines based on active directory group membership.

As Apache Guacamole is a standard web application, the communication has been secured by utilizing HTTPS communication. The presence of proxy or corporate firewall does not prevent the use of Apache Guacamole. Two main parts of Apache Guacamole are Guacamole web application running in Apache Tomcat and Guacd i.e., Guacamole daemon. The web application is responsible for authentication and communication with end users' web browser. Guacd supports many remote protocols and establishes connections towards remote nodes.

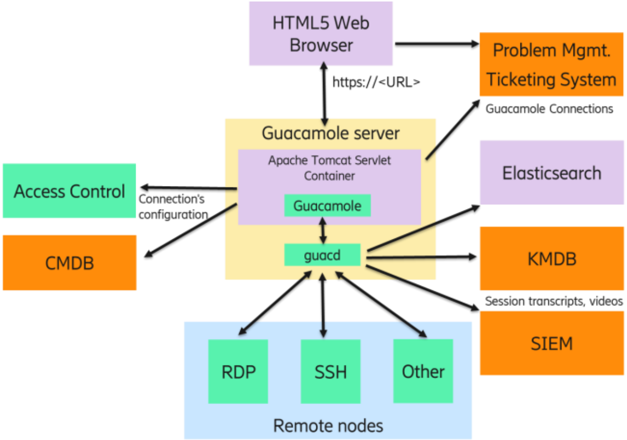

Figure 3. OWL architecture – traffic flows and integration points

B. *Data Collection*

One of the features that Apache Guacamole has is the ability to record sessions, like users' remote access session to the end node. By default, in OWL concept, all sessions have been recorded regardless of the type and this includes both, textual and GUI sessions.

The benefit over the traditional audit logging is that entire session is being captured from a user's login to the end node until remote access sessions ends. The captured session includes jumping from host to host and working inside applications. Also, this includes logging into devices that are not configured or cannot be configured to use the corporate authentication and where all users are connecting by using the common account, i.e., root.

A session transcript is extracted from the recorded session. Extraction from text-based sessions is a straightforward process, but it is more difficult for GUI-based session. For GUI-based session, the recording is coded into video stream and thus optical character recognition (OCR) needs to be performed on video frames to extract the transcript. The extracted transcript is enriched with metadata such as username, remote host name, session start time, etc., and is automatically stored into the Elasticsearch database for easier searching and manipulating at a later stage (Figure 3.) Recorded sessions are then converted to MP4 video stream compatible with streaming through modern web browsers.

By using Elasticsearch modules OWL toolset supports the ability to do the text search of the recorded video sessions. Fast search using free text or field-based search, filtering and reporting capabilities are provided by Elasticsearch's Kibana module. Kibana can be used to visualize, explore, and report Apache Guacamole sessions stored in Elasticsearch database. Those features can result in additional benefits if integration is done with ITSM system and the operational assurance tools.

C. *ITSM Integration*

As OWL toolset is based on ITIL best practices, one of the key integration points is ITSM integration which is a bridge between ITIL service transition and service operations activities. OWL integration would impact following ITSM components (Figure 3.):

- Configuration management database (CMDB)
- Problem and ticket management
- Knowledge management database (KMDB)
- Security information and event management (SIEM).

CMDB is normally used in ITSM as database to store information about configuration items (CI), such as hardware and software assets. Hardware type, operating system, IP address and other common IT information are stored as CI attributes in CMDB. In OWL toolset, Apache Guacamole connection information details to hardware assets are stored as connections' configurations, which enables easier access control. Since the Apache Guacamole connection is basically an URL link, this will give ability to any application that uses CMDB to extract the Apache Guacamole remote connection configuration information.

The ITSM problem and ticket management system module normally read data from CMDB to get the information about IT assets to which a ticket is being referred. With OWL toolset this process includes the information about the remote connection through Apache Guacamole. In this way, when a ticket arrives for some remote node, the ticket data includes a direct link that enables the remote connection to the affected end node. The operation team who is handling the ticket has an option to click on the link to access the affected remote node. In other words, the access to the remote node is a single click away and accessible through the ticket itself. Every session related to the open ticket can be automatically saved as KMDB item where recording session can be used as a "cookbook", for a competence build up or security actions reviews.

Integrated CMDB, KMDB, ticketing management and Apache Guacamole sessions data provide the information that answer who, what, when, where and how questions, with reference to the performed actions. The Apache Guacamole sessions can be used as a lesson learned and experience to increase operations assurance awareness and wisely improve ideas and judgment for further actions.

Traditionally, remote access is a different solution from a ticketing system and operations team needs to use separate tools and separate procedures to connect to remote machines mentioned in tickets. By integrating the ticketing system and remote access, the access to remote machines is greatly simplified, which shortens the response time.

D. ML/AI Control

Machine learning (ML) and artificial intelligence (AI) are key functionalities of today's cutting-edge technology which revolutionizes daily IT operations. ML/AI control is a logical step to take after data has been collected through different operations assurance tools within ITSM. This includes establishing data correlations with the standardized information to create mature knowledge based on which we can establish prediction and undertake ML/AI actions.

OWL toolset ensures automated actions, for example, possibility to receive automatic notifications or reports about predefined end node events or users' actions. This way security team can receive early threat detection in the remote node. ML/AI control will identify connections behind suspicions end users' actions, for example, copy or data transfer, and automated process will rapidly shut down access and connection eliminating threat.

Guacamole enables connection profiles for different groups, meaning that members of one group will be able to access only machines intended for that group. For example, access profile group for developers is different one for testers and different one for integrators. From session recording logs, it is possible to extract commands being executed by this group on sets of hosts or remote environments. Recorded data have been sent to ML/AI algorithm to be able to create correlation between group of users working on remote hosts and command being executed there. Different groups of users execute different application and have different ways of working. This would create a "way of working" model for each group of users. Using this model, it would be possible to differentiate normal user behavior from irregular one and from potentially dangerous one. In this case ML/AI control would raise an alarm and send events to SIEM for further investigation.

IV. PROOF OF CONCEPT

The following components have been used in proof of concept (PoC):

- Apache guacamole
- Microsoft Active Directory
- Elasticsearch

Apache Guacamole was configured to enable access to machines in one of the test environments located at Ericsson Nikola Tesla lab. A mix of connection types was configured, for example SSH text-based connections, as well as Remote Desktop (RDP) GUI connections. Guacamole was setup to use corporate Active Directory server for authentication. Authorization was handled by Active Directory group membership. This way, only authorized users with a specific group membership are allowed to access remote connections through Guacamole. Depending on group membership, users are presented only with remote connection associated with these groups. This way it is possible to segregate access to remote resources depending on Active Directory group membership.

After user session has finished, a batch process is initiated to extract session transcript from recorded session. For text-based sessions, this is simple sessions recording available in text format. For GUI-based session process is more complicated and an OCR has been performed on video frames. For OCR engine Tesseract [9] open-source OCR solution has been implemented. Performing OCR on screen recording does not provide clean output. Due to a nature of how modern GUI looks i.e., multiple overlapping windows, icons and other desktop elements, OCR of such picture will result in unorganized text. This text is difficult to read on its own as resulting text is combination of many desktop elements.

Resulting text obtained from OCR operation together with its video frame timestamp is stored as index to Elasticsearch. Besides these, additional fields are stored as well such as username and hostname on which session was running. This makes easy to search through many connection sessions using Kibana dashboard. As each Kibana search result returns both searched text and link to video frame from which this exact text was recovered, it makes it easy to search even through OCR obtained text which is difficult to read on its own.

The OWL has passed proof of concept (PoC) after its implementation in a selected project where OWL users had diversified target audience, such as integration teams, operational assurance teams or security teams.

In this case, the focus areas of the proof of concept were the solution stability, performance, and usability in daily business. The solution stability has a great impact on operational assurance teams, like level 1 to level 3 operations support teams, who need to act quickly with reference to any end node incident or issue. Integration teams are also heavily dependent on stability as all end node installations or upgrades require seamless and fast access connection.

Due to diversified target audience, the solution's performance required additional monitoring. OWL toolset can be run on standalone servers or virtual machines. In case of Apache Guacamole, it can be run as a docker container. During the PoC session CPU and RAM utilization, as well as local disk utilization were monitored. Long video sessions can increase the size of the recording, which may require additional local storage resources. The proper sizing of all OWL components requires thorough analysis of the following data: the number of concurrent users, ITSM integration points definition and data retention policy.

Sizes of generated video files of recorded sessions vary greatly. A recording's size depends both on the session's duration and on frequency of screen changes.

Video encoding and OCR recognition are CPU intensive tasks, where recommendation is to dedicate a separate virtual machine for video recording processing.

The use of OWL solution in a DevOps project team's daily business demonstrated its major benefits. During the development phase of project delivery, many issues where captured and stored as a session. For example, a session was captured during which a solution integrator (SI) had the task to configure database with required user privileges on different database schemas. The solution integrator missed to setup privileges rights to one set of data. Every project delivery has security auditors which review project delivery, in this case end node (database) security setup. After the performed penetration test, the security auditor (SA) noticed that he managed to enter unprotected database. The security auditor used Elasticsearch to search database integration and he found a missed step for database privileged access setup, after which the solution integrator fixed the problem. The key benefit was an opportunity to fix the problem before the project went into production and before the handover to operations team.

In this way, the coordinated action of the solution integrator and the security auditor, with simple use of the OWL toolset, prevented possible attacks and denial of service. Development and operations team agreed that actions such as penetration test, search for vulnerabilities and quick fixes can be automated and performed during end node development and integration process. The same process can be captured as evidence during the handover from development to operations team before end node goes into production environment.

V. CONCLUSION

Operations wisdom logging (OWL) toolset is currently being developed for the multiple production environments. The integration with different operational assurance tools and information technology service management (ITSM) is in the focus of the development. The proof of concept obtained very positive end users' feedback which put special emphasis on the fact that OWL offers one single platform with consolidated data, supporting one-click access to end node. Optical character recognition (OCR) enables excellent queries searching engine, providing easy browsing of remote sessions, or captured knowledge. The added value for users is that OWL becomes a central operational point which reduces the number of tools, browsers, different accesses and helps teams to reduce their response time.

As operational assurance in daily business and node handling are key target areas for OWL users, the solution has significant impact on the remote access process simplification, and it significantly reduces response time. Automated captured knowledge gathered during actions preformed increases troubleshooting efficiency and helps to consolidate scattered logs and knowledge in one single output ready to be consumed.

As OWL components are FOSS based, the solution has a very high commercial potential, starting with very low total cost of ownership (TCO). The OWL solution is vendor agnostic and can be implemented on end customer premises, common data centers, clouds or any other suitable environment required by clients. The OWL toolset aims to improve end customers' experience of DIKW process by fully utilizing ML and AI capabilities, as each OWL user can create his own profile based on daily business needs and automatically execute predefined actions. This way user or group profile can be a reference input for model preparation based on ML lifecycle [10]. OWL solution lifecycle facilitates complete access and data collection process transforming them to the repeatable process which makes correlations within CMDB and IPCC and prepares ML model for execution in daily operations.

REFERENCES

[1] Free and Open-source software, Available at: https://en.wikipedia.org/wiki/Free_and_open-source_software#CITEREFFeller2005, [Accessed: 07-Feb-2022]

[2] G. Nilsson: Open Source Software Within Ericsson Group, Ericsson internal document, Stockholm:2013.

[3] Apache Guacamole software, Available at: Implementation and architecture — Apache Guacamole Manual v1.4.0, [Accessed: 07-Feb-2022]

[4] Office of Government Commerce (OGC), ITIL - Service Strategy, LAG, 2007, ISBN 9780113310456.

[5] Office of Government Commerce (OGC), ITIL - Service Design, TSO, 2007, ISBN 9780113310470.

[6] Office of Government Commerce (OGC), ITIL - Service Transition, LAG, 2007, 9780113310487.

[7] Office of Government Commerce (OGC), ITIL - Service Operation, LAG, 2007, 9780113310463.

[8] Office of Government Commerce (OGC), ITIL – Continual Service Improvement, LAG, 2007, 9780113310494.

[9] Tesseract OCR engine - https://github.com/tesseract-ocr/tesseract, [Accessed: 07-Feb-2022]

[10] ML Lifecycle, Javapoint, Available at: Life cycle of Machine Learning - Javatpoint, [Accessed: 07-Feb-2022]