Improving Security of Oil Pipeline SCADA Systems Using Service-Oriented Architectures

Nary Subramanian

Department of Computer Science
The University of Texas at Tyler
3900 University Blvd.
Tyler, Texas 75799, USA
nsubramanian@uttyler.edu

Abstract. Oil pipeline Supervisory Control and Data Acquisition (SCADA) systems monitor and help control pipes transporting both crude and refined petroleum products. Typical SCADA system architectures focus on centralized data collection and control – however, this system has vulnerabilities that decrease the overall security of the system, especially for an oil pipeline SCADA. Service-oriented architecture (SOA) helps to improve security of SCADA systems by providing more localized data collection and control. In this paper we describe an SOA-based architecture for oil pipeline SCADA system that provides improved security compared to traditional architectures. An SOA-based SCADA divides the entire length of the pipeline system into zones where services offered within a zone are controlled by the zone master and masters periodically batch-update the central database over the back-bone network. The feasibility is explored by mathematical analysis and emulation.

Keywords: SCADA, petroleum, pipeline, architecture, services, security.

1 Introduction

Crude oil is terrestrially distributed by pipelines: from drilling rigs to crude oil storage tanks, from storage tanks to refineries, and finally the refined oil from refineries to gasoline storage tanks. Typically these pipelines span several thousands of miles – the US alone has about 150,000 miles of pipelines for transporting petroleum products [1]. In order to efficiently monitor and control this huge oil pipeline network supervisory control and data acquisition (SCADA) systems are employed. The oil pipeline SCADA has several hundred RTU's (remote terminal units) [14] that are connected to field instruments that measure pressure, temperature, and rate of flow of the oil flowing through the pipes, as well as change the statuses of valves and pumps along the pipeline. The RTU's communicate with a central master station using communication links such as satellite, cable, cellular, or fiber optic transmission media. The system architecture for traditional SCADA system is shown in Figure 1. A typical installation has several hundred RTU's communicating over dedicated links to a central master station [10, 11]. The most important aspect of oil pipeline is security [2, 3, 4, 5, 6, 7] and therefore SCADA systems are designed to provide real-time security status of the entire pipeline so that necessary action may be taken by the human agents monitoring the central information.

R. Meersman, Z. Tari, and P. Herrero (Eds.): OTM 2008 Workshops, LNCS 5333, pp. 344–353, 2008.

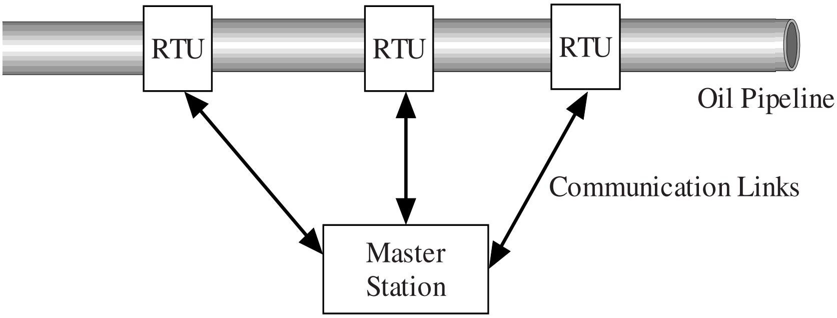

Fig. 1. Typical Oil Pipeline SCADA System Architecture

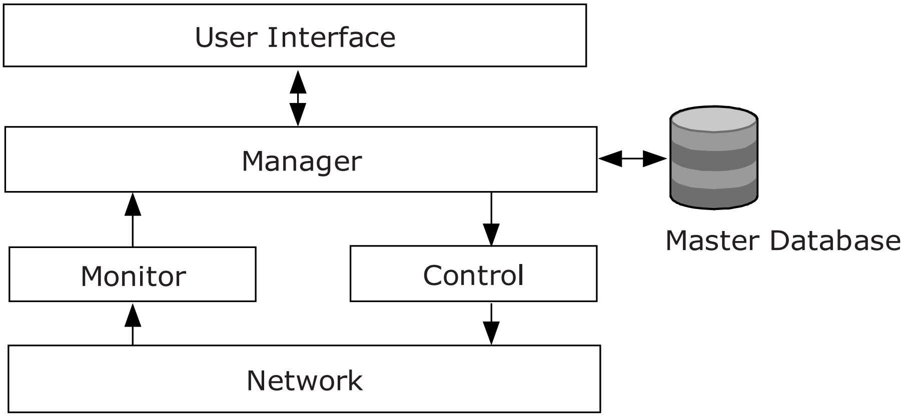

Fig. 2. Typical Software Architecture for SCADA System

Traditionally, software architecture of a SCADA system is a three layer architecture where the bottom layer is the data layer, the middle layer is the processing layer, and the top layer is the user interface layer. The layered software architecture for the SCADA system is shown in Figure 2. The processing layer accesses data from all RTU's regarding the status of various sensors and controls, and issues commands to the controls to change their states. The data received from the sensors and controls are stored by the processing layer in the data layer; besides, this data is also sent to the user interface layer for display to humans. Based on human responses to the data display, the user interface layer instructs the processing layer to change statuses of specific controls upon which the processing layer issues the appropriate commands to the relevant RTU's. This three layer architecture software resides in the master station of the SCADA system and all RTU's are assumed to be slaves in the system that send messages to and receive commands from the master. Therefore, the entire operation of the SCADA system is dependent on the network that connects the RTU's with the master. Oil pipeline SCADA systems communicate over several hundreds to thousands of miles and therefore need wide-area networking or the Internet to support their operations [10, 11]. Even though basic authentication mechanisms exist, security in oil pipeline SCADA systems are almost exclusively related to network security and several recent security breaches [5, 8] have occurred through the network. Therefore, the following main techniques have been suggested to improve oil pipeline security:

1. reduce network traffic: as discussed in [6, 7, 8] network is perhaps the most important component in a modern SCADA system from a security viewpoint. Therefore reducing network traffic will help improve SCADA security.
2. include people along the pipeline route in the security strategy: one of the latest strategies to improve oil pipeline security is to include local communities along the oil pipeline for the latter's operation and maintenance [3]. By allowing local communities develop a sense of ownership in the pipeline, the security of the pipeline improves.
3. avoid centralized control so that there is no one vulnerable critical point: the master station tends to be one of the main facilities of oil pipeline SCADA that serves as a vulnerable critical point. As pointed out in [15] such vulnerabilities need to be removed to improve security of pipeline installations.

In this paper we propose a new software architecture for oil pipeline SCADA systems that employs the concept of services, divides the entire length of the system into zones, and several zones may be collected into groups. Software using service-oriented architecture (SOA) [12] now runs in the processing layer of each zone and each zone master controls only that zone: this significantly reduces long-distance network traffic. Moreover, each zone is controlled by people from that area and this encourages local people to take ownership in the operation and security of the pipeline. Periodically, zones may send information to their group master; however, there can now be any number of group masters and this avoids having one centralized master station; moreover, masters may be dynamically reconfigured. The major advantages are improved security, improved reliability by avoiding single point of failure and improved maintainability of the system by involving local businesses along the length of the pipeline. The feasibility of the SOA-based SCADA system for oil pipelines is explored using mathematical analysis and actual implementation.

This paper is organized as follows: Section 2 discusses SOA-based software architecture for oil pipeline SCADA, Section 3 discusses feasibility analysis of the SOA-based system, Section 4 presents our observations on the SOA-based system, and Section 5 presents our conclusions and possible directions for further research.

2 SOA-Based Software Architecture for Oil Pipeline SCADA

In the SOA-based system, the entire length of the pipeline is divided into several zones and there are several group masters. Each zone monitors and controls only its zone and therefore most of the data traffic is localized. Each group master monitors the status of all zones under its responsibility – the group master keeps track of the status of each zone under it: the status may be as simple as knowing the overall security of each zone or as complicated as completely replicating each zones' user interface in detail. Group master is not a separate workstation but one of the zones taking on the responsibility of being the group master. In the trivial case, there is only one

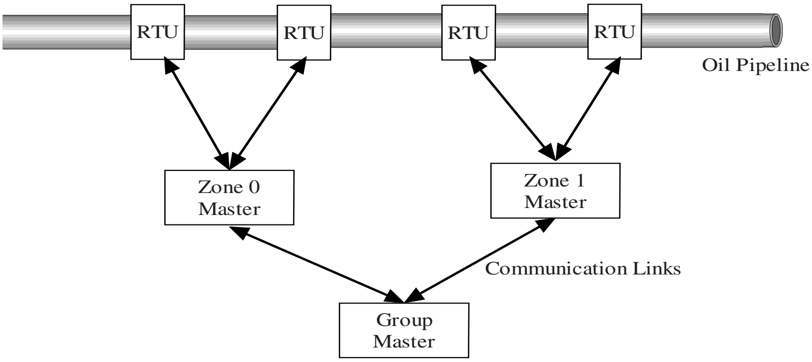

Fig. 3. System Architecture for Oil Pipeline SCADA Employing SOA

zone and that zone is also the group master – this corresponds to the traditional SCADA configuration. Each zone belongs to only one group master and each group master can have arbitrary number of zones assigned to it. The system architecture for SOA-based SCADA is shown in Figure 3.

The software architecture for the SOA-based system is shown in Figure 4. Each zone has its own user interface, manager, and database. In addition, each zone has its own web service broker and web services directory. All services for monitoring of RTU's, control of RTU's, and network access are registered in the web services directory and the web service broker accesses these services whenever needed. Likewise each zone registers its status interface with the group web services directory over the backbone network and the group master accesses the status of each of the zones assigned to it using these interfaces. The group master logs the details of its interactions with the zones on its database.

The SOA-based configuration significantly reduces communication requirements. The distances are now localized within each zone the size of which is set based on the needs of a specific system (a zone could be a critical portion of the system, a state, a province, a country, a geographic region, or the entire system itself), and the only inter-zone data transfer is that of the status of each zone. As described in the validation section of this paper, mathematically it can be shown that for normal inter-zonal data the communication requirements are reduced by about 75% for a four-zone system.

The SOA-based configuration actively encourages local businesses to participate in the oil pipeline security management by allowing them to register their services such as alternate network access or data analysis with the zonal web services directory that the zone manager can access if needed. As discussed in [12] only trusted businesses are allowed access to the web service broker for registration purposes and the security of the system will not be compromised by this procedure.

The SOA-based configuration also allows dynamic reconfiguration so that the role of the group master may be assigned to any of the zone masters. For this purpose each zone registers its interfaces with each other and this permits a zone master to access the status of each zone assigned to it. If a zone gets affected so that no communication is possible (or the zone must be shut-off for some reason), the group masters can quickly reassign zones between themselves for easier control of the entire system.

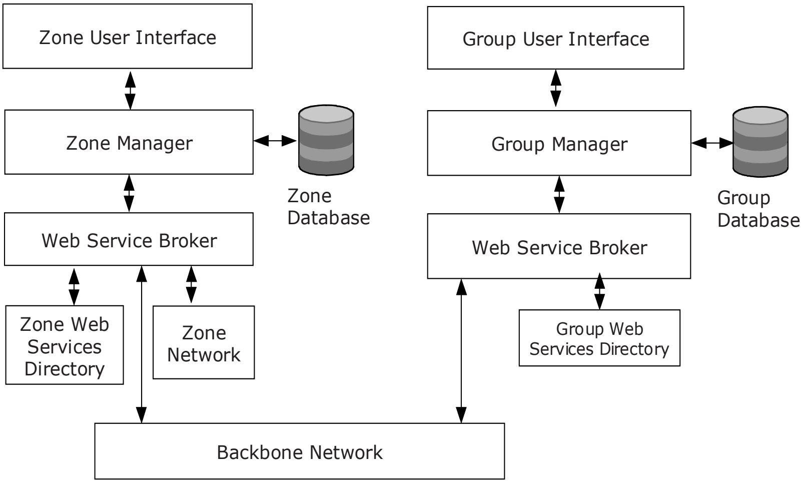

Fig. 4. Software Architecture for SOA-Based Oil Pipeline SCADA

3 Feasibility Analysis of SOA-Based SCADA System

Mathematical analysis of the communication requirements for a SOA-based oil-pipeline SCADA is given below. For the purposes of this analysis we assume that the oil-pipeline is 1000 miles long and there is an RTU for every mile (this assumption is validated by practice since RTU's are usually uniformly distributed along the length of the pipeline [2]). We assume each RTU makes six measurements.

For traditional SCADA configuration, a central master, located 500 miles from either end of the pipeline,

Total data communication requirement = 2 x 6[1 + 2 + 3 + … + 500] reading-mile
= 1,503,000 reading-mile
= 12,024,000 bit-mile,

assuming one reading takes a byte.

For a four zone SOA-based configuration, with zones distributed uniformly, that is, each zone is responsible for 250 miles of the pipe length,

Zone data communication requirement = 2 x 6[1 + 2 + 3 + … + 125] reading-mile
= 756,000 bit-mile

Total zonal data communication requirement = 4 x 756,000 = 3,024,000 bit-mile.

If each zone master updates a group master that is represented, for the purposes of this discussion, by a hypothetical master in the middle of the pipeline, then if each zonal update takes 1 byte, then

Zone data update communication requirement = 2[375 x 8 + 125 x 8] bit-mile
= 8000 bit-mile

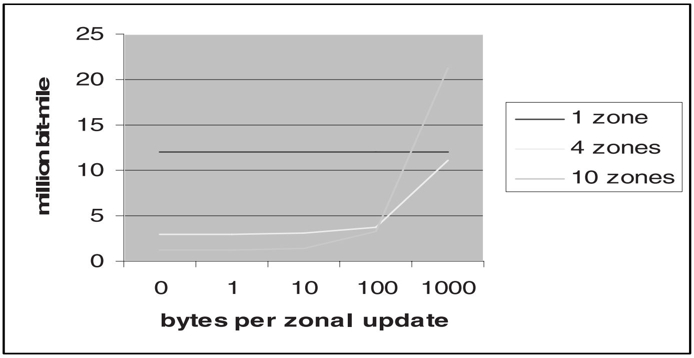

Fig. 5. Data Communication Requirements for Varying Zonal Update Sizes

Total data communication requirement for an SOA-based system that takes 1 byte to update the group master is: 3,024,000 bit-mile + 8000 bit-mile = 3,032,000 bit-mile.

Similar calculations were performed for 10 byte, 100 byte, and 1000 byte zonal updates for both 4 zones and 10 zones, and the results are shown in Figure 5.

As can be seen in Figure 5, 1 zone (equivalent to the traditional SCADA architecture) requires far more data communication requirements for normal zonal update data (< 1000 bytes): almost 75% more data communication requirement is needed by the traditional SCADA architecture.

We developed a physical implementation of an SOA-based system that emulated an oil-pipeline SCADA. This system transferred 1 byte per update and the group user interface is shown in Figure 6. As can be seen Figure 6 is sufficient for displaying the status of each of the four zones – zone that is not working efficiently is displayed in different color. This group interface conveys sufficient information to the human agent for monitoring at a high level each of the four zones.

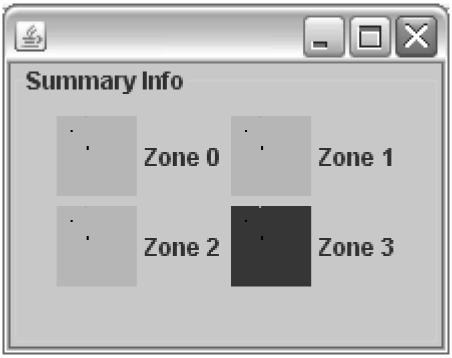

Fig. 6. Group User Interface Displaying Summary Zone Information

4 Observations

4.1 SOA-Based Architecture is Better than Distributed Architecture

The SOA-based SCADA for oil pipeline is a better alternative than a distributed architecture: in a distributed architecture the components are distributed but the interfaces are hard-coded. This not only does not help local businesses to easily provide their services but also does not help dynamic reconfiguration.

4.2 Improved Security

All the three requirements for improved security, namely, reduced communication requirements, participation of local communities, and decentralized master are satisfied by the SOA-based architecture. Communication requirements are only a fraction (25%) of the traditional SCADA architecture, local businesses can easily integrate their services with the architecture, and masters can be dynamically reconfigured.

4.3 Improved Reliability

Reliability of the system increases because of removal of bottlenecks – if one communication link fails, local businesses can be called upon to provide alternate communication means to the zone and group masters, and if one zone were to be blocked out, then communication with other zones is still possible. Since each zone takes responsibility for RTU's under its control, focus is more detailed and therefore, overall reliability improves.

4.4 Improved Maintenance

Maintenance is assigned to local businesses and therefore it is more timely and effective. Likewise, if any additional feature is required, local businesses can provide them as services that helps improve maintainability of the system.

4.5 Future Proofing

Recent technological advances such as VOIP (voice over IP) and instant messaging [9] are expected to boost ease of communication and ability to monitor and control SCADA systems. However, with SOA-based system, it is easy to incorporate the new and other future developments by simply providing an interface to these technologies as services. The services may even be outsourced to local businesses to hasten adoption.

The SOA-based software architecture may be adapted to any pipeline SCADA system such as water, natural gas, and sewage transmission systems. However, cost incurred in developing a distributed system using SOA needs to be weighed against potential benefits. In the case of oil pipelines, with the current cost per barrel of oil being in excess of $140 [13], and with systems typically capable of pumping one million barrels per day [3] through oil pipelines, the cost of oil transmitted per day is in excess of $140 million. For such valuable commodity, the cost of SOA-based system will be more than compensated by any potential losses due to security breaches.

5 Related Work

Oil conveying pipelines are part of critical infrastructure and SCADA systems are used extensively for monitoring and controlling the pipelines [3, 4]. Among the more important quality requirements of SCADA systems are security, reliability, and maintainability [16, 17, 19]. While several techniques have been proposed to improve security in SCADA systems – for example, redundancy [16] has been used to improve security for communication networks, intranet-based technology for real-time control to improve reliability and maintainability has been proposed in [18], and specific countermeasures for a set of vulnerabilities for electrical systems has been proposed in [19]. Detailed reliability analysis for SCADA systems have been performed in [20]. However, it has been suggested that SOA can be used in safety critical environments [21], and in this paper we considered SOA for improving security for oil pipeline SCADA systems – however, the use of SOA for SCADA necessitates a zone-based strategy so that geo-political interests are satisfied as well since pipelines frequently traverse national boundaries. The SOA-based approach offers promise to improve security, reliability, and maintainability of oil-pipeline infrastructure.

6 Conclusion

Supervisory control and data acquisition (SCADA) systems for oil pipelines monitor and control transfer of oil and petroleum products through the pipeline over several hundreds to thousands of miles. This is typically accomplished by having a central master station communicating over a variety of communication links with several hundred remote terminal units (RTU's) to monitor various physical parameters and to control valves and pumps along the pipeline to keep the oil flowing through the pipeline. Among the most important requirements to improve the pipeline security is to reduce network traffic, include local communities along pipeline route in the security strategy, and to avoid vulnerabilities such as having only a single master station. In this paper we propose an service-oriented architecture (SOA) based SCADA system for oil pipelines that helps to improve security, reliability, and maintenance. Our proposal includes a modified system architecture that divides the length of the pipeline into zones and groups where a group consists of several zones but one zone belongs to only one group. Each zone has a zone master and one of the several zone masters in a group also becomes the group master. By using services, the service brokers at the zone and group level and able to identify interfaces in other zones to form a dynamically reconfigurable architecture. The SOA based approach reduces network traffic, provides ability to local businesses to participate in the pipeline processes, and avoids vulnerabilities associated with having only one master. The feasibility of the SOA-based oil pipeline SCADA architecture was explored mathematically and by physical implementation. The network requirements for SOA-based system are typically only about 25% of the traditional SCADA architectures.

For the future we need to validate the system with more robust group master user interface that provides more details of the status of the zones and allows control of each zone as well. We also need to validate the dynamic reconfiguration of the group master by physical implementation and/or simulation. Another future activity is to validate the SOA-based architecture when RTU's capture more readings as well as when RTU's are non-uniformly distributed. Also, the distinction between security and safety need to be delineated so that they are separately addressed. Moreover, it has been suggested that SOA is inefficient for security purposes [22] and that SOAP XML messages actually increase traffic size [23] – both these aspects need further investigation from an oil-pipeline SCADA standpoint. However, we believe that SOA-based SCADA is a promising and profitable option for improving oil-pipeline security.

Acknowledgements. The author wishes to thank the anonymous reviewers of the earlier version of this paper for their detailed and thorough comments.

References

1. http://en.wikipedia.org/wiki/List_of_countries_by_total_length_of_pipelines (accessed on July 5, 2008)
2. http://en.wikipedia.org/wiki/Pipeline_transport (accessed on July 5, 2008)
3. Ismailzade, F.: A Strategic Approach to Pipeline Security. Report of the Institute for the Analysis of Global Security (2004), http://www.iags.org/n1115043.htm
4. Clementson, D.P.: Reviewing SCADA basics. Pipeline and Gas Technology (2006) (accessed on July 5, 2008), http://www.pipelineandgastechnology.com/story.php?storyfile=2defd4c7-bdad-4776-af94-fa948ad21b18.html
5. Slay, J., Sitnikova, E.: Developing SCADA Systems Security Course within a Systems Engineering Program. In: Proceedings of the 12th Colloquium for Information Systems Security Education, pp. 101–108 (2008)
6. Idaho National Engineering and Environmental Laboratory. A Comparison of Oil and Gas Segment Cyber Security Standards. Report No. INEEL/EXT-04-02462 (2004)
7. API Standard 1164, Pipeline SCADA Security First Edition (September 2004) (accessed on July 5, 2008), http://api-ep.api.org/filelibrary/1164PA.pdf
8. Sauver, J.: SCADA Security (accessed on July 5, 2008), http://darkwing.uoregon.edu/~joe/scada/
9. Henrie, M.: API 1164 Standard Revision (accessed on July 5, 2008), http://www.api.org/meetings/topics/pipeline/upload/Morgan_Henrie_API_1164_Standard_Revision_API_Presentation_REv_1.pdf
10. Press Release, Sinopec selects Invensys for SCADA system on China's longest crude oil pipeline (October 11, 2005) (accessed on July 5, 2008), http://news.thomasnet.com/companystory/468291
11. References for Cegelec installations (accessed on July 5, 2008), http://www.oilandgas.cegelec.com/References/ScadaRef.htm
12. O'Neill, M., et al.: Web Services Security. McGraw-Hill, New York (2003)
13. Associated Press news report, Oil passes, settles above $145 for first time (July 3, 2008) (accessed on July 5, 2008), http://www.msnbc.msn.com/id/12400801/
14. http://en.wikipedia.org/wiki/SCADA (accessed on July 5, 2008)
15. Matheson, M., Cooper, B.S.: Security Planning and Preparedness in the Oil Pipeline Industry. In: The Oil & Gas Review, pp. 104–108 (2004)

16. Farris, J.J., Nicol, D.M.: Evaluation of Secure Peer-to-Peer Overlay Routing for Survivable SCADA Systems. In: Proceedings of the 36th Conference on Winter Simulation, pp. 300–308. ACM Press, Washington (2004)
17. National Communications System, Technical Information Bulletin 04-1, Supervisory Control and Data Acquisition (SCADA) Systems (October 2004) (accessed on August 23, 2008), http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf
18. Ebata, Y., Hayashi, H., Hasegawa, Y., Komatsu, S., Suzuki, K.: Development of the Intranet-based SCADA (supervisory control and data acquisition system) for Power System. In: IEEE Power Engineering Society Winter Meeting, vol. 3, pp. 1656–1661. IEEE Press, Los Alamitos (2000)
19. Dagle, J.E., Widergren, S.E., Johnson, J.M.: Enhancing the Security of Supervisory Control and Data Acquisition Systems: the Lifeblood of Modern Energy Infrastructures. In: IEEE Power Engineering Society Winter Meeting, vol. 1, p. 635. IEEE Press, Los Alamitos (2002)
20. Bruce, A.G.: Reliability analysis of electric utility SCADA systems. IEEE Transactions on Power Systems 13(3), 844–849 (1998)
21. Prinz, J., Kampichler, W., Haindl, B.: Service Orietned Communication Architectures in Safety Critical Environments. In: Integrated Communications Navigation and Surveillance (ICNS) Conference (2006) (accessed on August 23, 2008), http://spacecome.grc.nasa.gov/icnsconf/docs/2006/04_Session_A3/06-Kampichler.pdf
22. Roch, E.: SOA Security Architecture (2006) (accessed on August 23, 2008), http://it.toolbox.com/blogs/the-soa-blog/soa-security-architecture-11431
23. Leonard, P.: High Performance SOA – A Contradiction in Terms? (2006) (accessed on August 23, 2008), http://www.webservices.org/weblog/patrick_leonard/high_performance_soa_a_contradiction_in_terms