An Autonomic Approach to Security Incident Response and Prevention in Cloud Computing

Glenn Russell, Roy Sterritt

Presented by **Glenn Russell**
School of Computing
Faculty of Computing, Engineering
and the Built Environment
Ulster University
russell-g6@ulster.ac.uk

IARIA Congress 2023

Autonomic Computing Systemization of Knowledge Session

Glenn Russell

- MSc A.I. Candidate at Ulster University
- Autonomic Computing interest from MSc AI module: COM760 Autonomic Computing and Robotics

- I've spent over two decades building technical and management skills in everything from software development to cloud-native infrastructure. I'm continuously learning, and bringing my years of experience to bear on building and leading teams solving data engineering and cloud-native problems.

- Glenn Russell | LinkedIn https://www.linkedin.com/in/glenn-russell-aa08341b5/

Cloud Computing

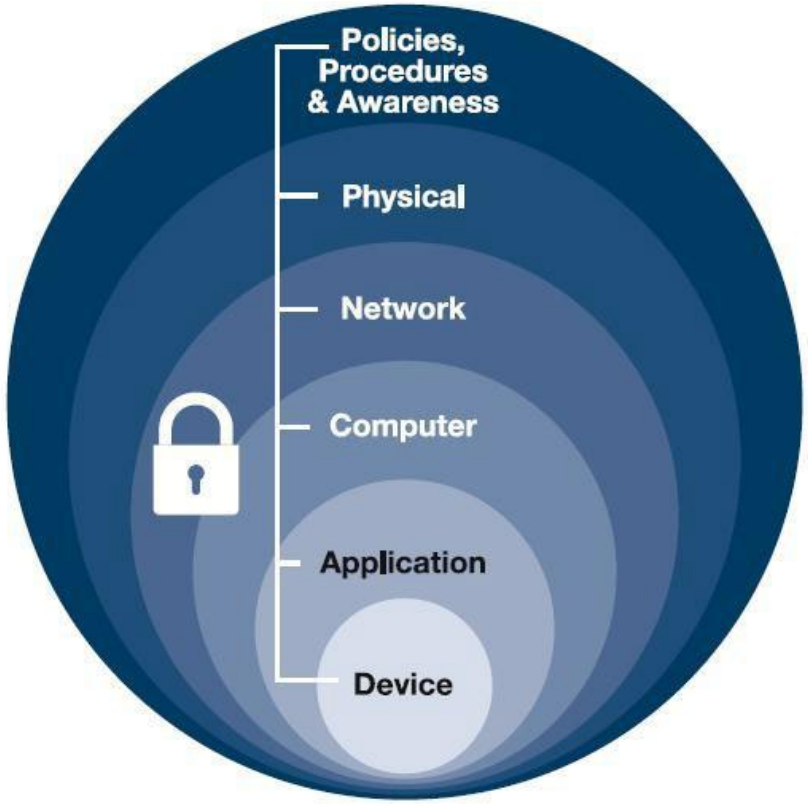
IBM lists the benefits as:

- **Flexibility** allows services to be accessed and scaled to fit ever changing demands, from anywhere on the internet.
- **Efficiency** means that users do not need to spend money on physical equipment, much of which may be redundant, while being able to bring applications to market quicker.
- **Strategic value** is derived from having access to the latest technology as it becomes available, from new processors to the latest machine learning platforms.

IBM, "Benefits of Cloud Computing," 10 October 2018. [Online]. Available: https://www.ibm.com/uk-en/cloud/learn/benefits-of-cloud-computing

Security in Cloud Computing

Regardless of the computing paradigm being deployed, effective security programmes adopt some fundamental principles which can be utilised regardless of the computing paradigm. The principle of these approaches is *defence in depth (figure)*, which calls for a series of defensive mechanisms which are layered to protect valuable data and information. Having multiple layers of security ensures that there are redundant controls in place if a specific control is compromised.

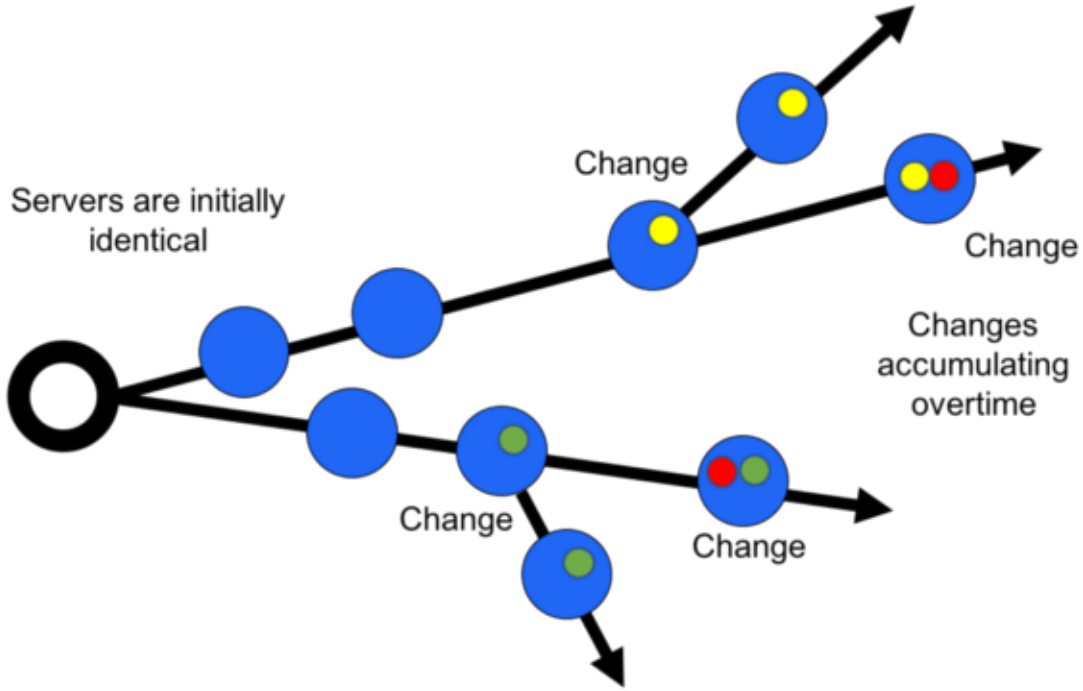


Forcepoint, "What is Defense in Depth?," Forcepoint, 2022. [Online]. Available: https://www.forcepoint.com/cyber-edu/defense-depth

Autonomic Cloud Computing and Security

Cloud Computing was Autonomic Computing's major impact success during its 2nd decade []. These principles are refined into just four, so-called *self-CHOP*.

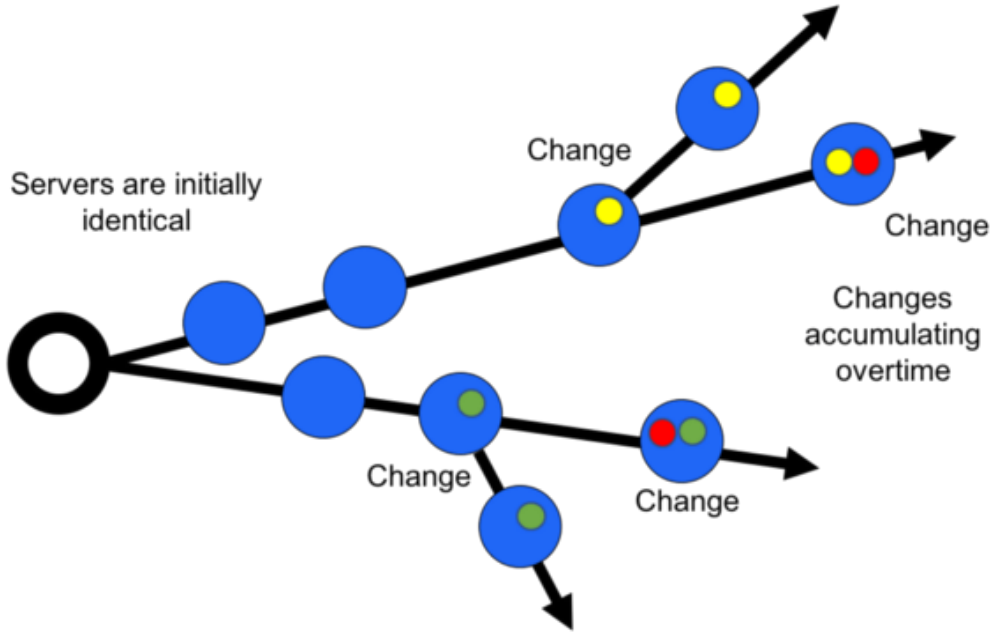
- Self-*configuration*
- Self-*healing*
- Self-*optimisation*
- Self-*protection*



[] R. Sterritt, "Keynote: 20 Years of Autonomic Computing," in International Conference on Autonomic and Autonomous Systems (ICAS), Online (Covid-19), 2021.

Autonomic Cloud Computing and Security

Popular tools in this space are *Puppet*, *Chef*, *Ansible* and *Terraform*. Each of these tools possess a management component or master which configures new and existing components such as new servers coming online via an agent or surrogate agent process. Both the master and agents are akin to an *autonomic manager*, which exchange information on the desired state of an environment and the actual state. *Configuration drift* (figure) is where the configuration of a service differs from the expected configuration, and it is this that the master attempts to correct for each service. It does so iteratively through a process called *eventual consistency*, in which the master issues commands over a secure channel to make corrections, and the services (in fact, an autonomic manager) respond with a snapshot of their current state. This continues until there is no configuration drift. This mechanism results in what is a *self-healing* process that can operate with any aspect of the cloud that can be managed programmatically.



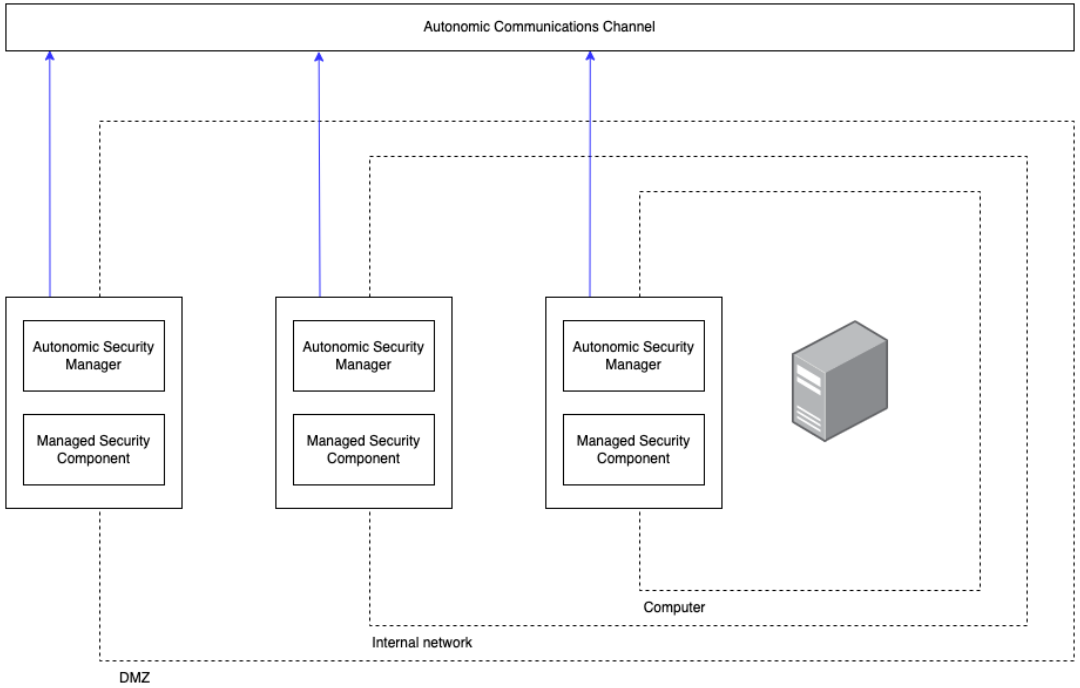
Security platforms such as IBM QRadar Risk Manager [] disabled features which allowed the automatic configuration of security controls for these reasons.

[] IBM, "IBM QRadar Risk Manager," 24 January 2022. [Online]. Available: https://www.ibm.com/docs/en/qsip/7.3.2?topic=manager-qradar-risk

An Autonomic Incident Response System

An autonomic solution requires that the following components be present in the solution:

- An **autonomic element** which is a combination of a managed component and an autonomic manager
- The **managed component** which in this case could be any kind of security apparatus that we would to managed autonomically, e.g., a firewall or user access list
- The **autonomic manager**, which operates the managed component based on feedback such as messages received from the environment.
- Communication between the autonomic elements will be achieved with an **autonomic communications channel**. As part of this, messages will be formatted according to open standards such as STIX, CyBOX and SCAP.

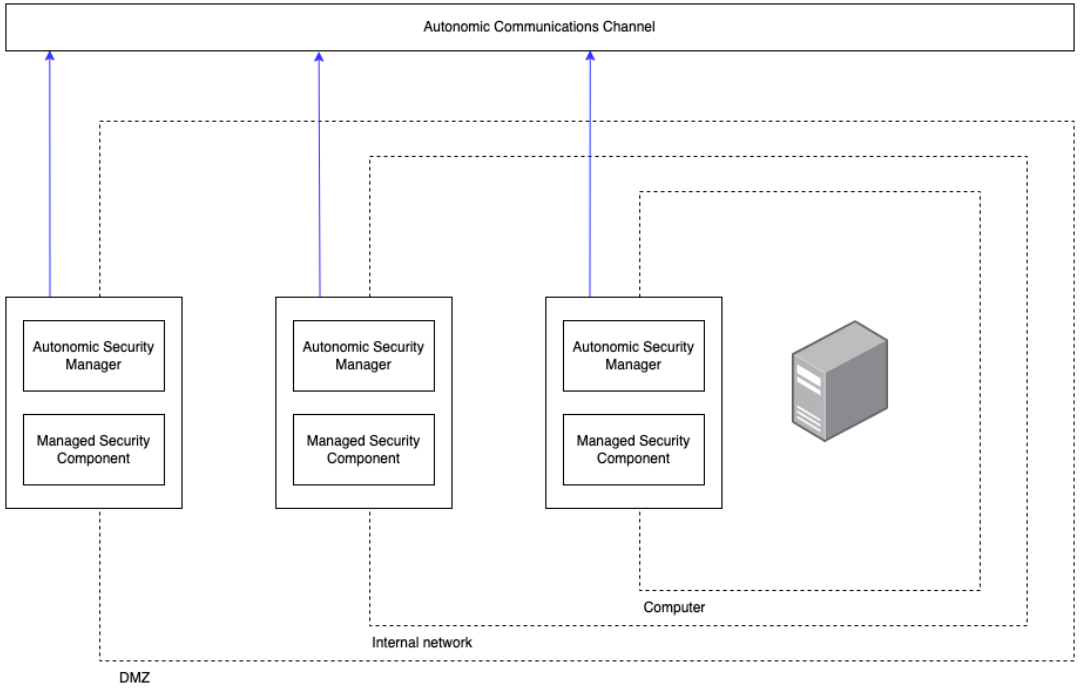


An Autonomic Incident Response System

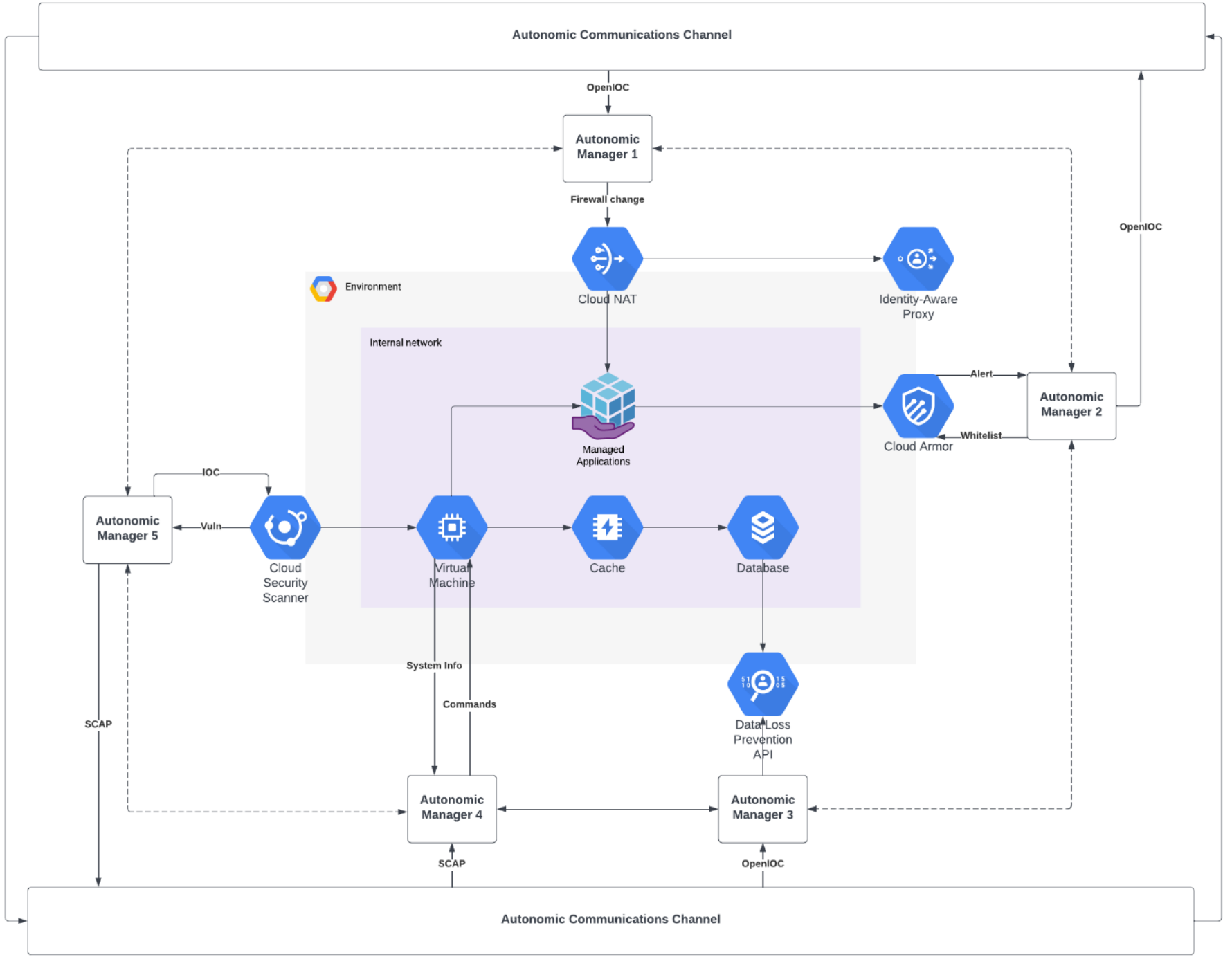
The **environment** should be considered as the full extent of a cloud deployment which hosts infrastructure that provides some value, of any combination of services. In the case of a website, this could be virtual machines, databases, message queues and an in-memory cache, for example. A defence in-depth strategy (figure) calls for multiple layers of security. The solution proposes that each logical layer of such a strategy is secured by an autonomic element as described.

Autonomic elements must consume and emit the following kinds of messages:

- **Indicator of compromise**
- **Vulnerabilities**
- **Mitigations**

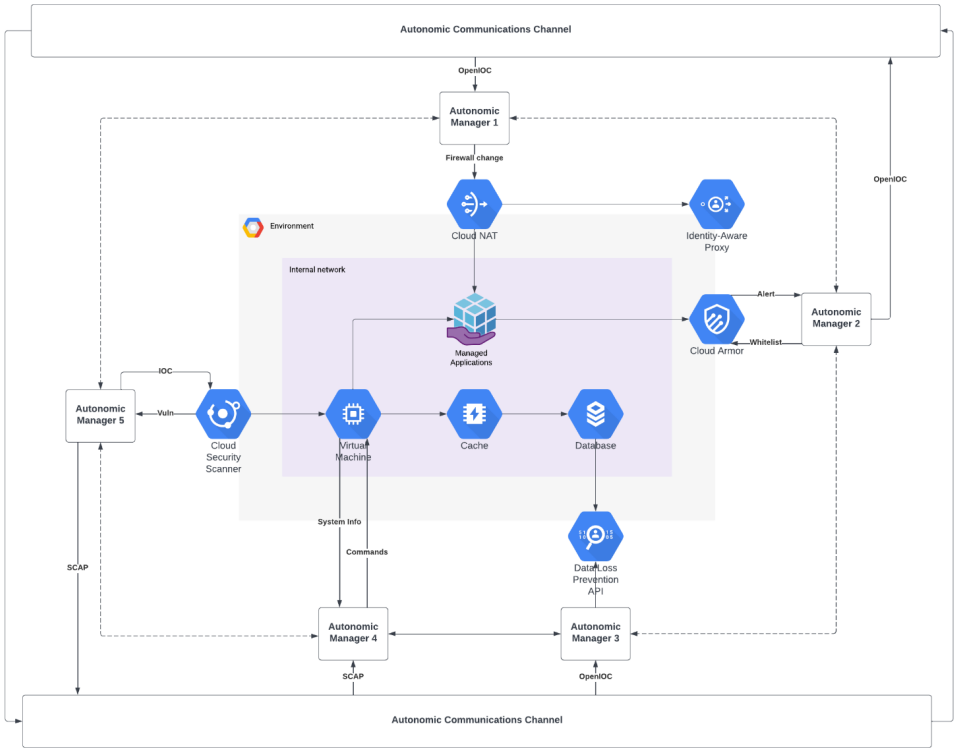


Autonomic secure cloud environment



An Autonomic Incident Response System

- A scalable and fault tolerant message bus such as Kafka or RabbitMQ will constitute the **autonomic communications channel** which each autonomic manager will both subscribe and publish to. These message queues are built to ensure that messages are always delivered and can scale up to many millions of messages per second, so important security events are guaranteed to be delivered. Each managed component is an existing security control or cloud service. The autonomic manager integrates with it via existing APIs and acts as a gateway between control specific messages and the standardised formats the solution is relying on. While each autonomic element receives every reflex signal being triggered, it is up to each specific element to decide how to react to it, and if it also needs to transmit a reflex signal in turn. By combining both cloud services and security controls into a single autonomous system, an immune system is created which removes the need for a human in the loop because existing security tools integrate poorly with the environment they are protecting.

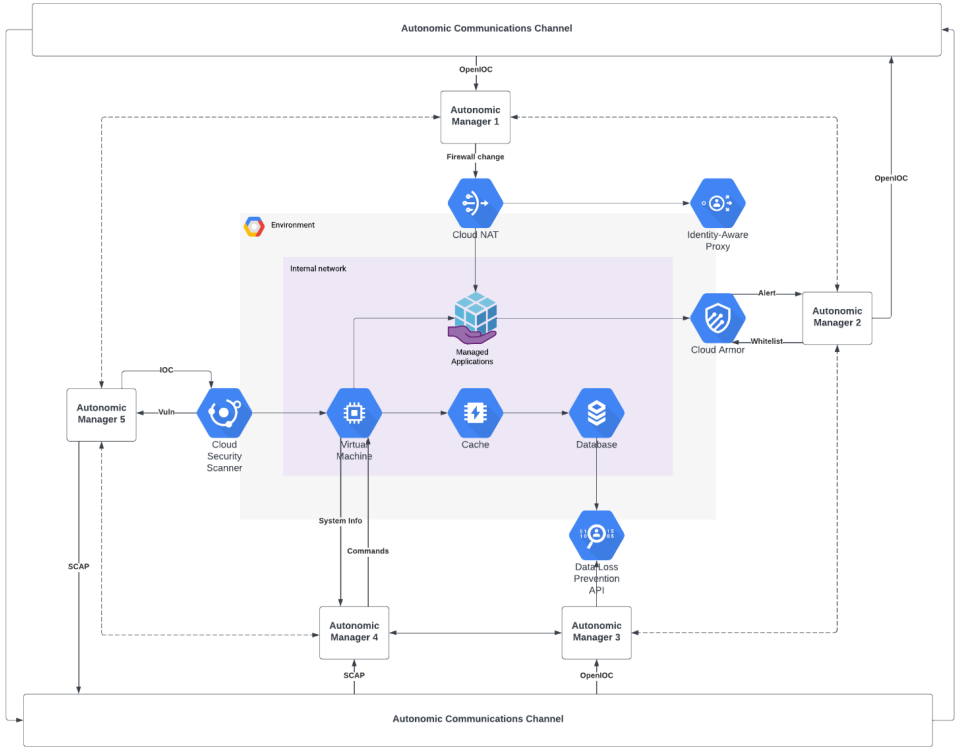


An Autonomic Incident Response System

• To understand how the solution would work, consider figure. AM5, which manages a vulnerability scanner, detects a vulnerability on a VM and emits an SCAP message with remedial details. AM4 receives the message and issues a system command which updates the environment state with the remedial action and the change is made as configuration drift has occurred between the desired state and actual state. AM1, AM2 and AM3 receives the message but does not perform any action.

• In another scenario, AM3 detects that data is being sent to an unauthorised IP outside of the environment. It emits an OpenIOC message. AM1 receives the message and instantly enacts a change to the environment to block this network traffic. In addition, AM2 receives the message and queries the Cloud Armor web application firewall (WAF) for all traffic sent from the offending external IP address and emits an OpenIOC message. Upon receiving the message AM5 conducts a vulnerability scan of the web applications being hosted that interacted with the external IP, based on the messages from AM2.

• At no point in these interactions is a human necessary to perform any action. This fact is the advantage of an autonomic security solution, as the workload on security practitioners is greatly reduced.



Conclusion

- So far, the security industry has failed to take advantage of the many features covered by this paper, and only increased the complexity of systems overall, failing to take advantage of autonomic principles in favour of artificial complexity.

- Taking a devil's advocate position; a serious ethical issue with the solution is the consumption of data which has historically been heavily biased against network traffic originating in certain regions, and weight that traffic is much more negatively based on this fact alone.

- The root cause of this is the bias in threat intelligence which is either directly or indirectly consumed by security tools and cloud platforms. This will directly lead to users from those areas being treated differently than others based on an explicit bias.

- However, the move to purely autonomous security platforms can greatly reduce this issue by removing very real cognitive bias introduced by human operators.

- So, in this case, a very real bias results in blind spots as teams may not consider 'friendly' nation states as potential sources of attack. This is underlined by the simple fact that attempting to search for material associated with nation state threat actors will yield results that are almost exclusively non-western countries.

- Finally, the ability for an autonomous system to operate purely on observations and data effectively negates this very real shortcoming in 'human-in-the-loop' security platforms. Autonomicity provides the opportunity to remove bias from the system along with its stated aim of intelligent self-management.

Acknowledgements

Glenn Russell managed development of the product in this presentation / paper for IBM.

This paper was produced as part of COM760 Autonomic Computing & Robotics on Ulster University's MSc in Artificial Intelligence.

Thank You

Presented by **Glenn Russell**
School of Computing
Faculty of Computing, Engineering
and the Built Environment
Ulster University
russell-g6@ulster.ac.uk

IARIA Congress 2023