**Team Project: Agent System in Digital Forensics**

**by Alberto Castro, Maksym Dudkin, Vasilisa Lukashevich**

**(Intelligent Agents May 2023)**

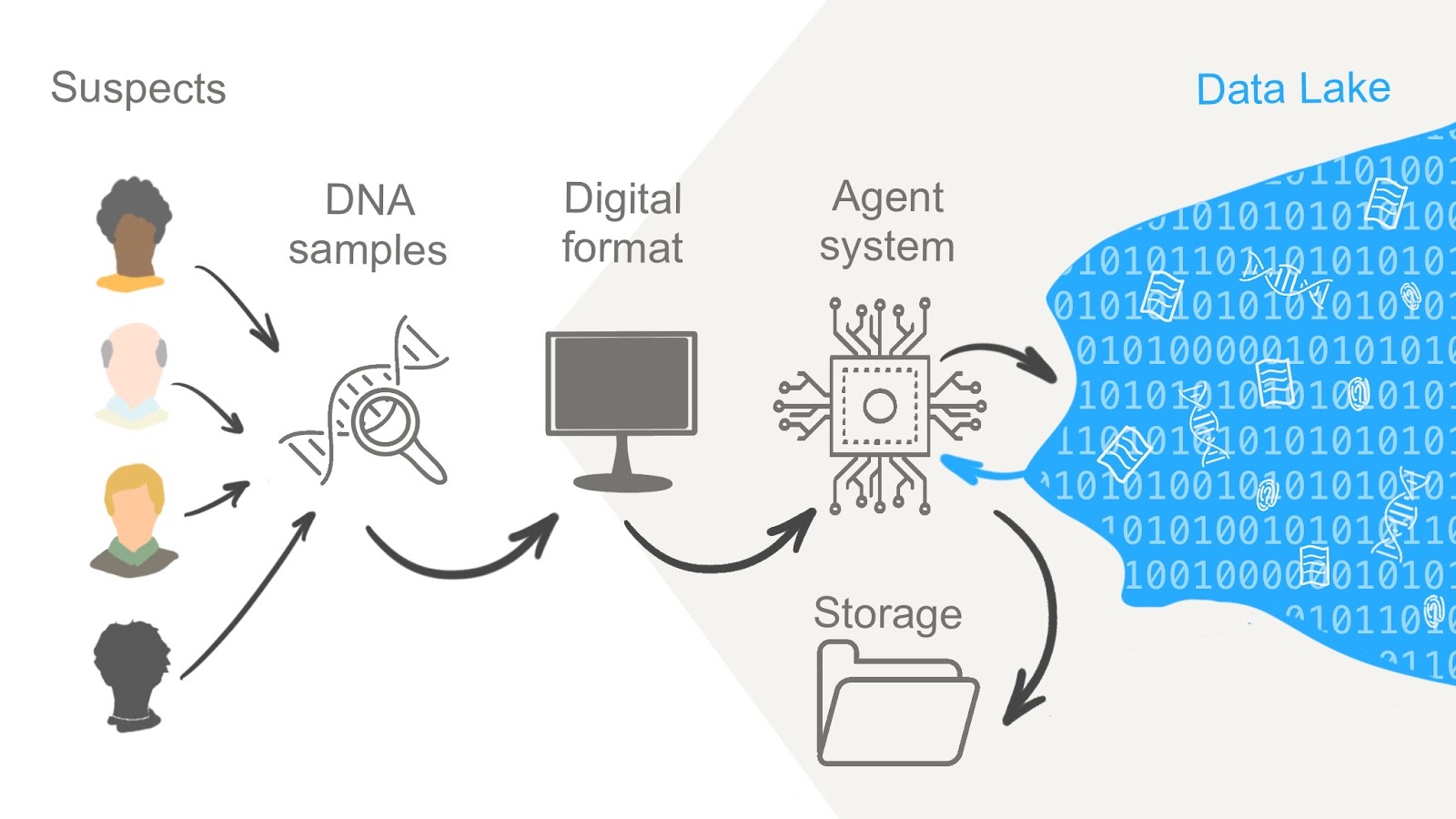
**Introduction**

In this proposal document, we provide a system for the Digital Forensics domain for a detective agency to find specific file types on a file system, archive them and send the results somewhere for further analysis. We propose to perform this task using an agent system.

Agents are independent parts of software capable of working together in multi-agent system to solve problems that could not be solved by a single agent. The main characteristics of intelligent agents are autonomy, reactivity, proactivity, adaptability, communication & coordination with other agents (Kendrick et al., 2016).

**The general idea**

In this case, we are attempting to find a DNA match for serial killer suspects by comparing their DNA samples with a DNA data lake of materials. For example, we aim to identify if the DNA matches those found on victims in a specific area over the last 20 years from the database for the cases still to be investigated.

****

**Proposed agent system architecture**

The system assumes inputs in a special format for the digital DNA samples. There is one main agent to work with these files and two modules.

**Coordinator agent**

According to (Terra, 2023) this decentralised agent is responsible for giving tasks to the monitor agents. It counts the files it needs to analyse, instantiates a finite number of monitor agents, and then indicates to each monitor agent what files it must pre-analyse to find suspicious files.

In our DNA example Coordinator agent is responsible for:

1. Searching for new packages

2. If found - open and analyse

3. Take a DNA sample and send it to the Analysis Module.

4. Get a response from the Analysis Module

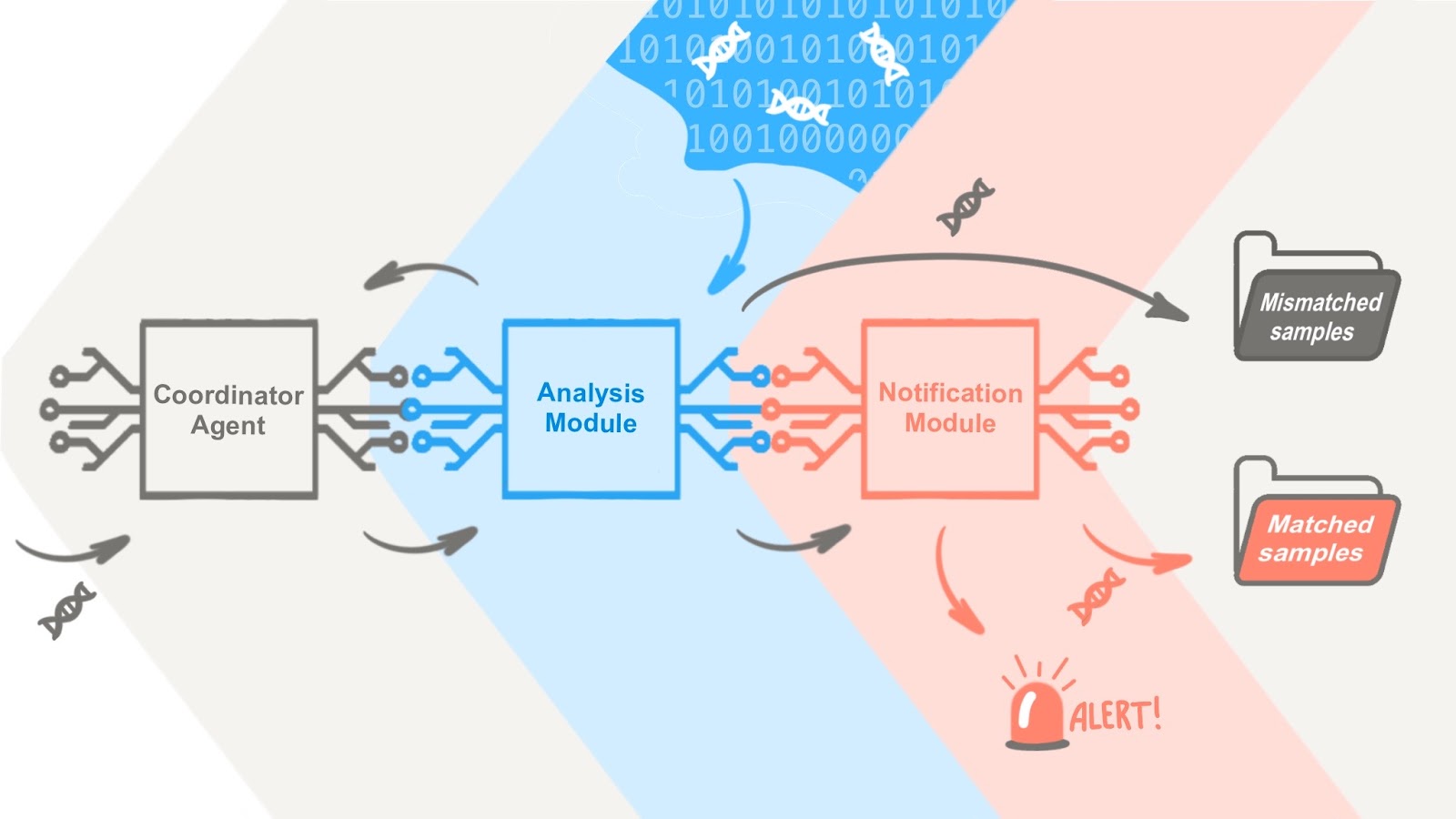
**Analysis module**

A Machine Learning module, which gets DNA sample and looks if the database has this specific DNA:

1. If it has: -> Get the case number related to the DNA sample -> Return the DNA sample and case numbers associated with the DNA in the package

-> The system creates an alert "The match!" in the detective or police systems with the additional **Notification module**.

2. If DNA or case has not been found -> Create a new non-criminal case DNA record in the special storage based on ethical principles.



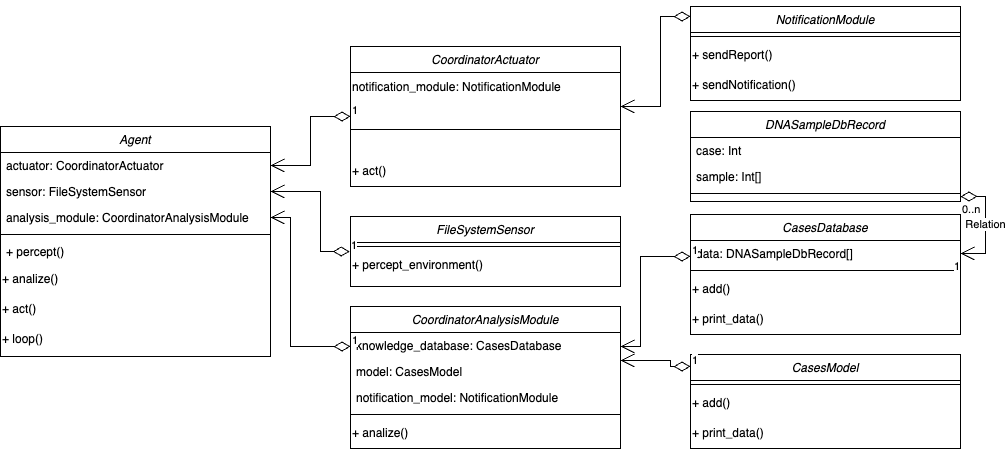
**Agent implementation**

We used Python to implement a basic prototype of an agent with different modules built-in. Agents' primary purpose is to find digitised DNA samples in files in a file system environment. Each DNA sample has the following format:

|  |
| --- |
| sample: {     case: Int,      dna\_sample: Int[]  } |

The *case* is a unique integer identifier, and each DNA sample is represented as an array of integers.

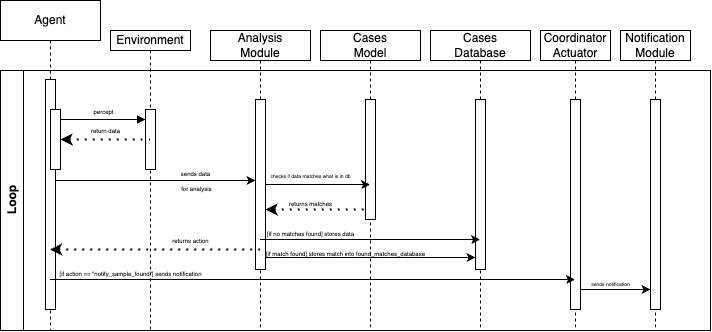
**Class diagram**



These are the classes listed in the class diagram:

* Agent: This is the agent class. It has three main properties: an actuator, a sensor, and an analysis module.
* CoordinatorActuator: This class triggers an action when a DNA sample is found. The only action defined so far is *notify\_sample\_found.*
* FileSystemSensor: This class is responsible for scanning and perceiving the environment looking for new data to pass to the agent.
* CoordinatorAnalysisModule: This class is responsible for analysing the environment and trying to find matches in the agent’s knowledge base.
* NotificationModule: This class sends notifications when a new DNA match occurs.
* DNASampleDbRecord: This data class models DNA samples in the database.
* CasesDatabase: This data class is used to model a database of DNA samples.
* CasesModel: This class contains the methods for matching DNA samples with samples stored in the database.

**Sequence diagram**

****

In the sequence diagram above, the agent’s behaviour is illustrated. The whole life cycle of the agent occurs within a loop. It means there is an infinite loop where an instance of the agent is fed indefinitely with DNA samples. The life cycle of an agent is described in the following list:

1. An instance of an agent is ready to scan its environment. This instance has access to a file system and can detect if there’s data in the system.
2. If the agent detects data from the environment, it reads that data and sends it to an internal analysis module to analyse it.
3. After that, the analysis module checks if the new sample it’s in its knowledge base. If not found, it stores the data in the knowledge base. If it's a match, it stores it in a different database where matches are stored.
4. The analysis module sends a response to the agent to let it know if a DNA sample was found.
5. If there is a DNA sample found, the agent sends a signal to an internal actuator that, at the same time, sends a signal to the notification module to send a notification about the new match found.
6. The cycle finishes, and the agent rereads data from the environment as in the first step.

**Conclusion**

Data privacy and security are some of the main challenges in developing applications that use sensitive data, such as biometric or medical data. In our case, we are using simulated data, but in a real-world scenario, we would need to use cryptographic techniques to ensure that sensitive data is safe.

Additionally, the development of intelligent agents involves some ethical concerns. According to Leslie (2019) and his FAST principles, systems of the nature of intelligent agents must be transparent, which means that they must be explainable to users and stakeholders. This is essential to foster trust and accountability in decisions made by intelligent agents, and it allows developers to address and correct potential biases or errors.

**References**

Bianchi, L. & Liò, P. (2007) Forensic DNA and bioinformatics. *Briefings in bioinformatics*, 8(2), pp.117-128.

Kendrick, P., Hussain, A.J. & Criado, N. (2016) Multi-agent systems for dynamic forensic investigation. In *Intelligent Computing Theories and Application: 12th International Conference, ICIC 2016, Lanzhou, China, August 2-5, 2016, Proceedings, Part I 12* (pp. 796-807). Springer International Publishing.

Leslie, D. (2019). Understanding artificial intelligence ethics and safety. The Alan Turing Institute. <https://doi.org/10.5281/zenodo.3240529>

Shtrauss, A. (2023) Biopython | Bioinformatics Basics. Available from: <https://www.kaggle.com/code/shtrausslearning/biopython-bioinformatics-basics> [Accessed 8 Jun 2023]

Terra, J. (2023) Exploring Intelligent Agents in Artificial Intelligence. AI & Machine Learning Course on SimpliLearn. Available from: <https://www.simplilearn.com/what-is-intelligent-agen-in-ai-types-function-article> [Accessed 10 Jun 2023]

Twardowski, B. & Ryzko, D. (2014) Multi-agent architecture for real-time big data processing. In 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) (Vol. 3, pp. 333-337). IEEE.