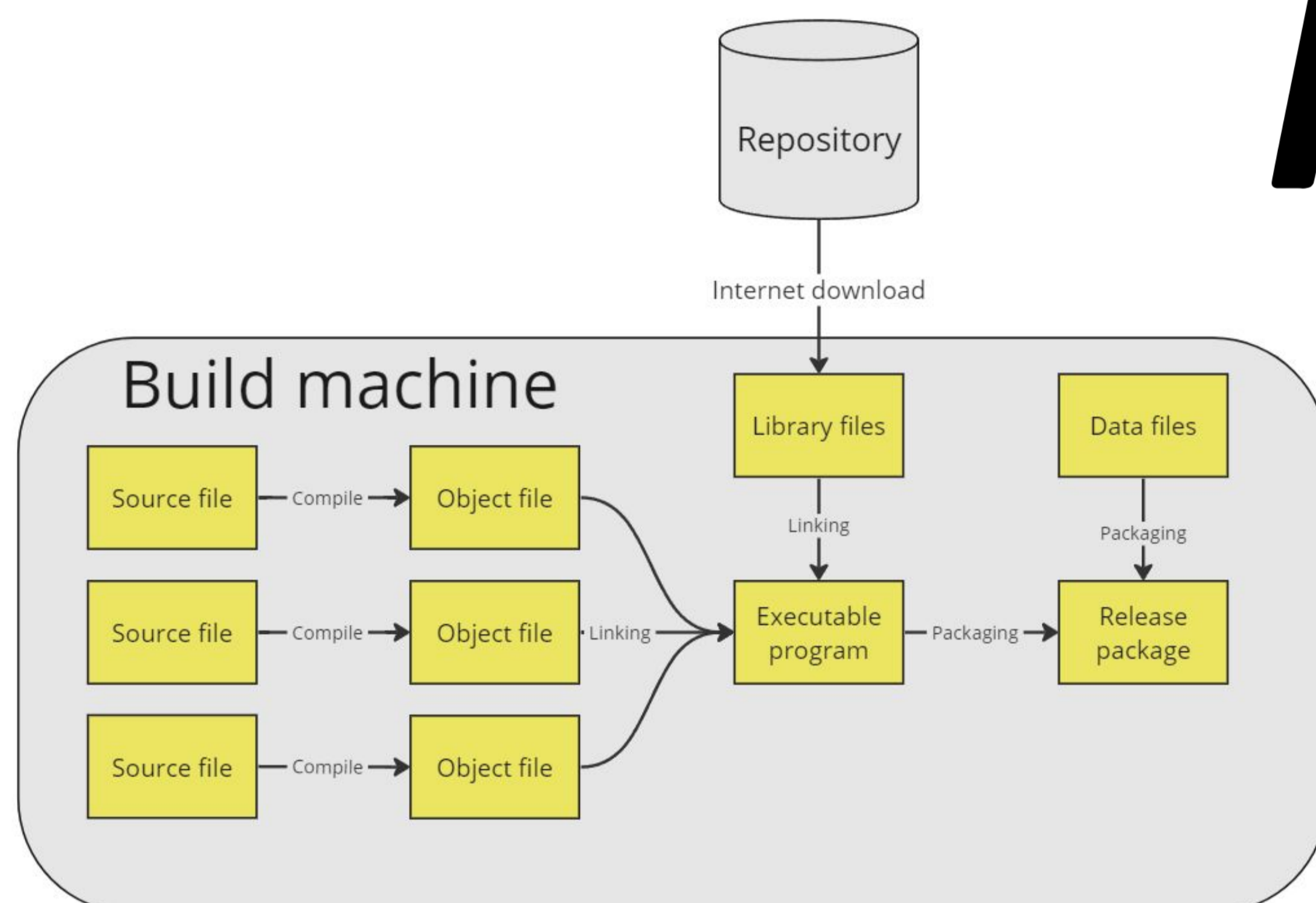


An Evaluation of Air-gapped Software Builds

Organizations which have a strong need for secure computer networks may employ a type of security measure where one or more computers are physically isolated from other computer networks such as nearby local networks or the wider Internet. This type of setup is known as an air-gapped computer network and is used in military facilities, financial institutions, medical devices, and industrial control systems. While it can be efficient in securing a computer network, it impedes the function of many software tools vital to the software development process such as many build automation tools which rely on Internet access.

In my thesis, I aim to propose an efficient solution to software builds within an air-gapped computer network by describing and evaluating a software architecture which enables efficient air-gapped builds.

Maven™



A normal build process using external dependencies.

But in an air-gap, the Internet download is not possible.

Is it as simple as downloading the dependencies outside the air-gap, placing them on a USB and giving it to any developer when they need it?

No, because you break many principles of DevSecOps! How do you:

- Ensure developers have easy access to the same versions of packages
- Ensure parity between different software environments (testing/production)
- Ensure a high level of security
- Update the air-gapped system safely and efficiently
- Allow a high degree of automation
- Scalability
- Tools integration
- How do you initialize tools like Maven inside the air-gap, which get their essential compile plugins from the Internet?