



Fabian Hauser



Raphael Zimmermann



Redbackup: A Redundant Distributed Backup System Prototype

Motivation

Today, most individuals and small to medium enterprises make backups on cloud environments or local storage media as e.g. hard disk drives or network attached storage systems (NAS).

These solutions require either high personal efforts to maintain local storage media or a high level of trust in a third party storage provider.

Currently, there are no backup systems available on the market which are both easy to use and provide the user with a high level of data security and privacy.

Results

The architecture consists of backup nodes, which store and distribute data directly over a network connection and a client application that creates and restores backups to or from nodes. Lastly, a management system is introduced to allow users to manage multiple backup nodes.

The presented prototype demonstrates the viability of our proposed architecture, introducing a reduced feature set. The prototype can create, distribute and restore unencrypted backups.

Project Goals, Approach and Technology

A backup system which solves this issues must not only provide a secure and reliable application to create and store backups, but also permit users without further domain knowledge to install and configure the application.

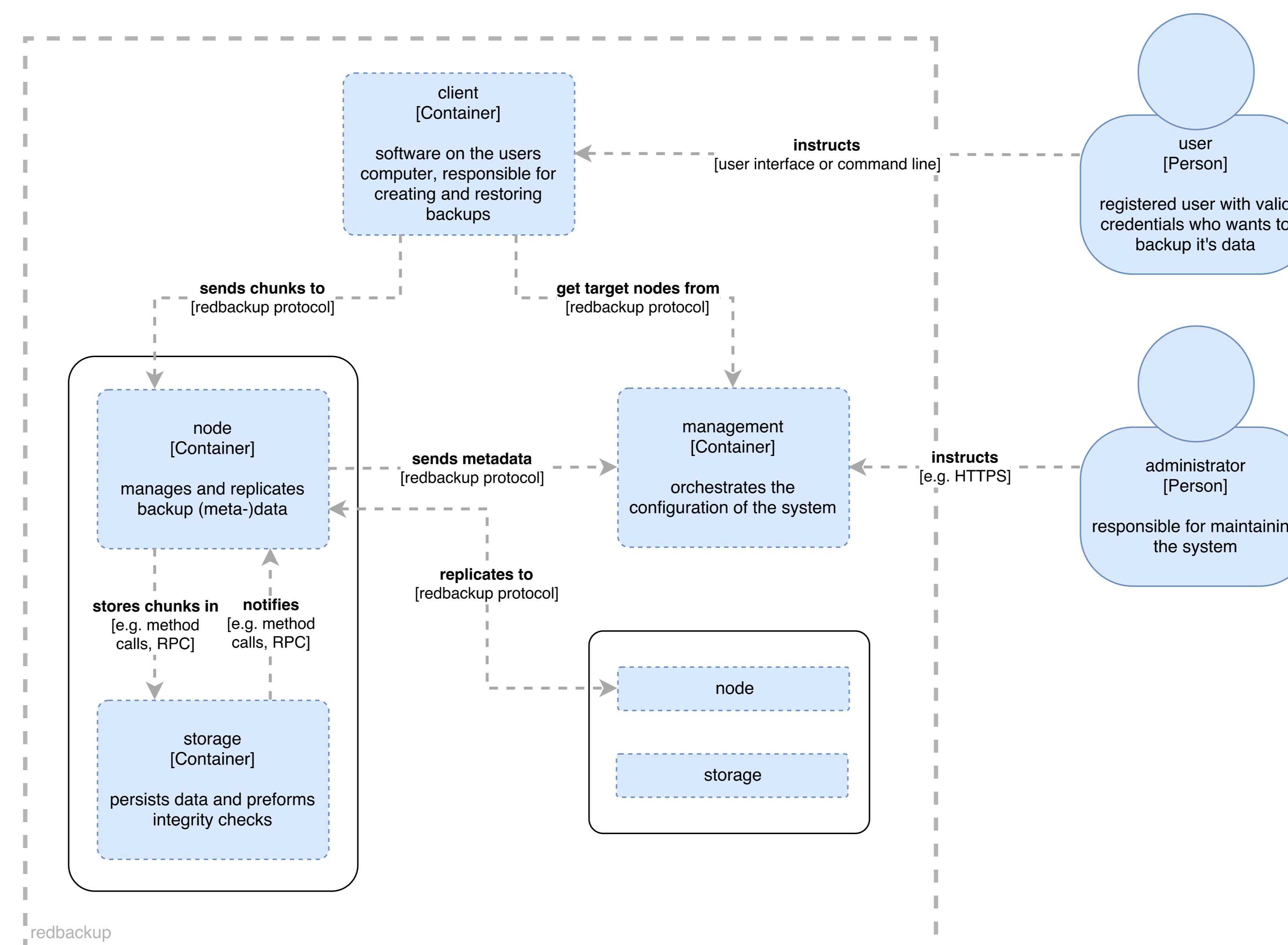
To meet this requirements, we further analysed and created a comprehensive architectural design.

The subsequent implementation of an architecture prototype took place in the *Rust* system programming language, which we learned during the course of this project. Rust enabled us to create a very stable yet efficient backup prototype.

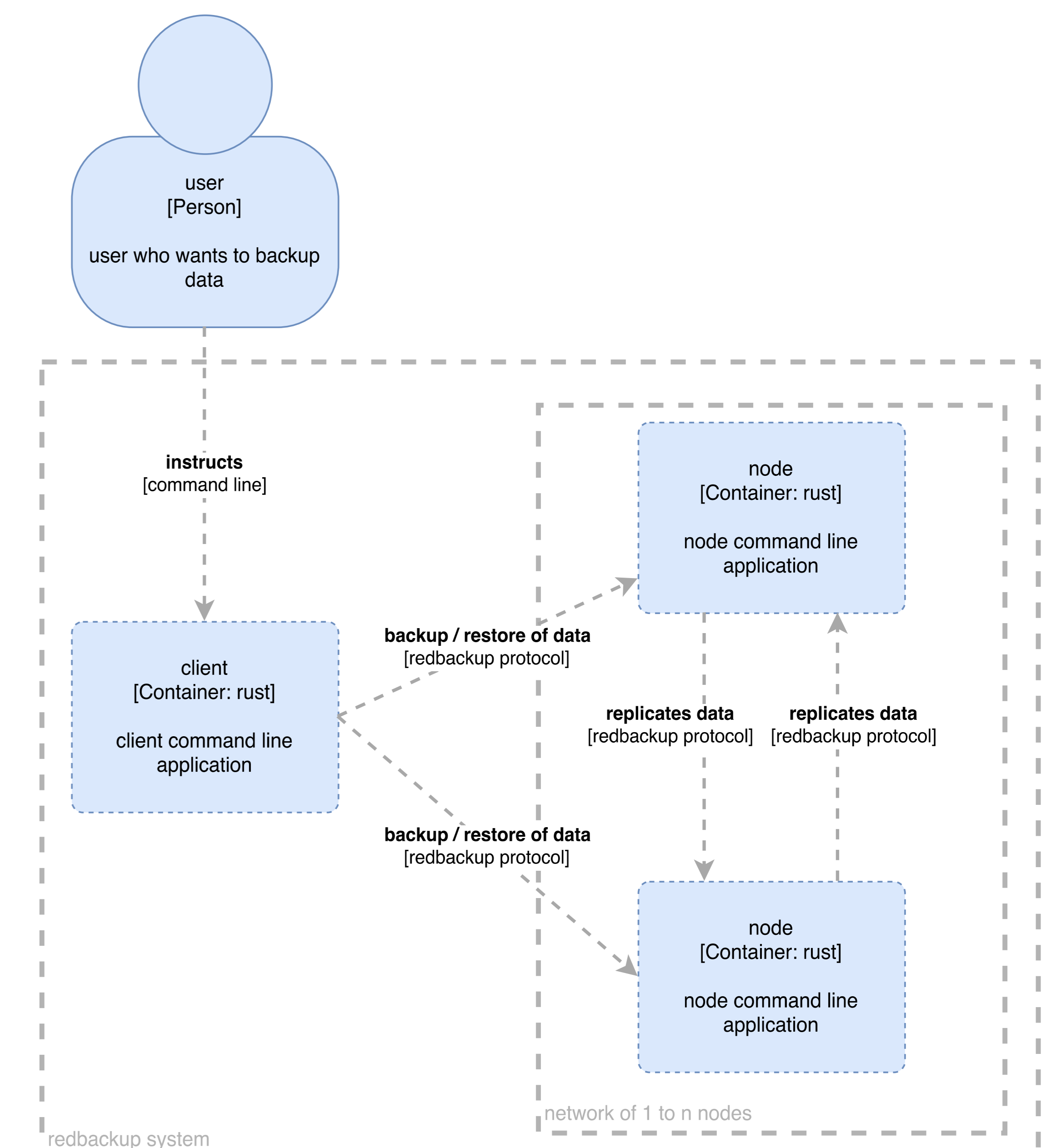
Prospects

To extend the prototype into a fully functional backup system, there are multiple functionalities and improvements that may be implemented. The main missing parts are backup encryption, splitting of backup data, advanced data distribution strategies and the management application.

With our prototype, we demonstrate the viability of the architecture and pave the way for further implementations.



C4 Container diagram illustrating the high-level shape of the redbackup software system and how responsibilities are distributed.



C4 Container diagram illustrating the components as implemented in the prototype