A Secure Selection and Filtering Mechanism for the Network Time Protocol

draft-ietf-ntp-chronos-04

Neta Rozen-Schiff, Danny Dolev, Tal Mizrahi, Michael Schapira

The Chronos Watchdog

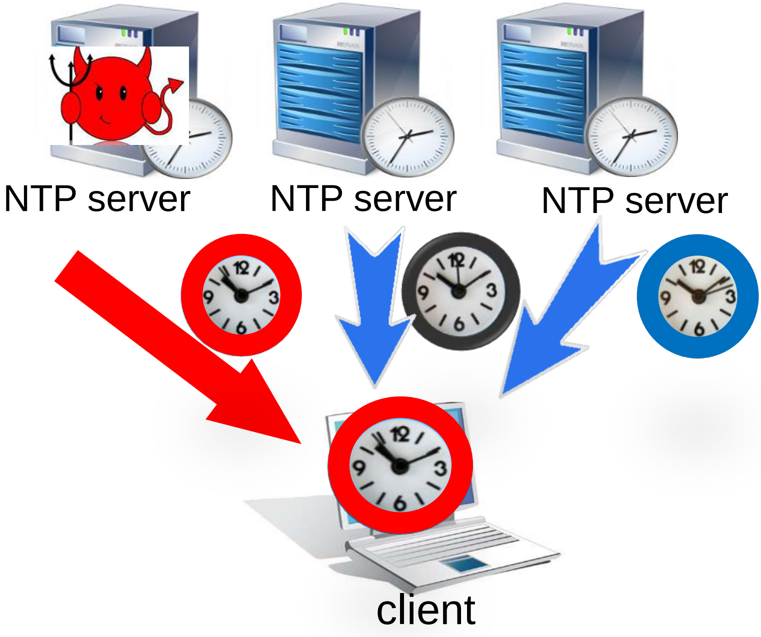
The Chronos Watchdog is **a security layer that wraps NTPv4’s (or NTPv5’s) time-computation logic**.

Chronos **protects the NTP client against time-shifting attacks** while **preserving the time accuracy and precision** of the default scheme.

Reminder: Threat Model

The attacker:

- Controls a large fraction of the NTP servers in the pool (say, ¼)
- Capable of both deciding the content of NTP responses **and** timing when responses arrive at the client
- Malicious



The Chronos Watchdog: Design Goals

The **Chronos NTP client** is designed to achieve the following:

- **Time accuracy and precision**
  - match the accuracy and precision of NTPv4 when **not** under attack
- **Provable security** in the face of fairly powerful MitM attacks
  - negligible probability for successful timeshifting attacks
- **Backwards-compatibility**
  - no changes to NTP servers
  - limited software changes to client
- **Low computational and communication overhead**
  - query few NTP servers

The Chronos Watchdog Architecture

- **Two concurrent modes:**
  - **Primary mode: NTPv4**
  - (Secure) **watchdog mode: Chronos**

- **Key idea:**

  Match NTPv4’s **accuracy and precision** by using NTPv4 to update the local clock when the client is **not under attack**. Significantly enhance **security** by using Chronos to update the local clock **when under attack.**

The Chronos Watchdog Architecture – cont.

- **Two different time scales to keep computation/communication overhead low:**
  - ➢NTPv4 updates at the same time granularity as today
  - ➢Less frequent Chronos time computations (e.g., once per 10 NTPv4 updates)
- Following each Chronos time computation, Chronos’ and NTPv4’s offsets are compared.

If the difference between NTPv4’s and Chronos’ offsets exceeds a threshold, an attack is detected, and Chronos’ offset is used to update the client's clock. Otherwise, NTPv4's offset is used.

Chronos Time Computation: Overview

Chronos’ design combines several ingredients:

- **Rely on many NTP servers**
    - Generate a large server pool (hundreds) per client
        - E.g., by repeatedly resolving NTP pool hostnames and storing returned IPs
    - Sets a very high threshold for a MitM attacker
- **Query few servers**
    - Randomly query a small fraction of the servers in the pool (e.g., 10-20)
    - Avoids overloading NTP servers
- **Smart filtering**
    - Remove outliers via a technique used in approximate agreement algorithms
    - Limits the MitM attacker’s ability to contaminate the chosen time samples

Chronos Time Computation: Components

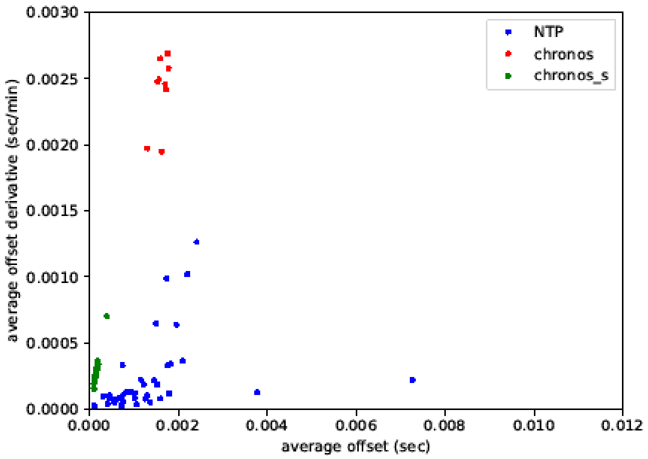
Chronos computation differs from NTPv4 in three key aspects :

- **Calibration process**
  - Generates a local pool of servers the client can synchronize with, consisting of n servers (up to hundreds).
- **Modified selection process**
  - Chronos relies on many NTP servers, chosen at random periodically
- **Modified cluster algorithm**
  - Chronos uses an approximate agreement technique to remove outliers

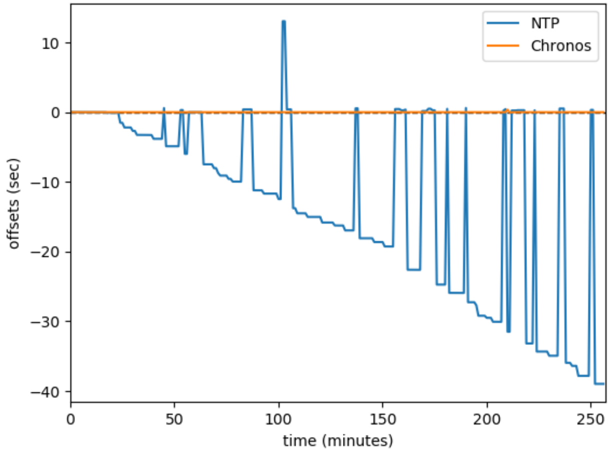
Chronos Time Computation *vs.* NTPd

- Chronos computation *vs.* NTPv4’s:
    - Greater variety of sampled servers over time
    - Avoids (NTPv4) source quality filters
    - Provable security guarantees
- Possible adverse effects on precision.

NTP *vs.* Chronos Offsets in Oregon (not under attack)



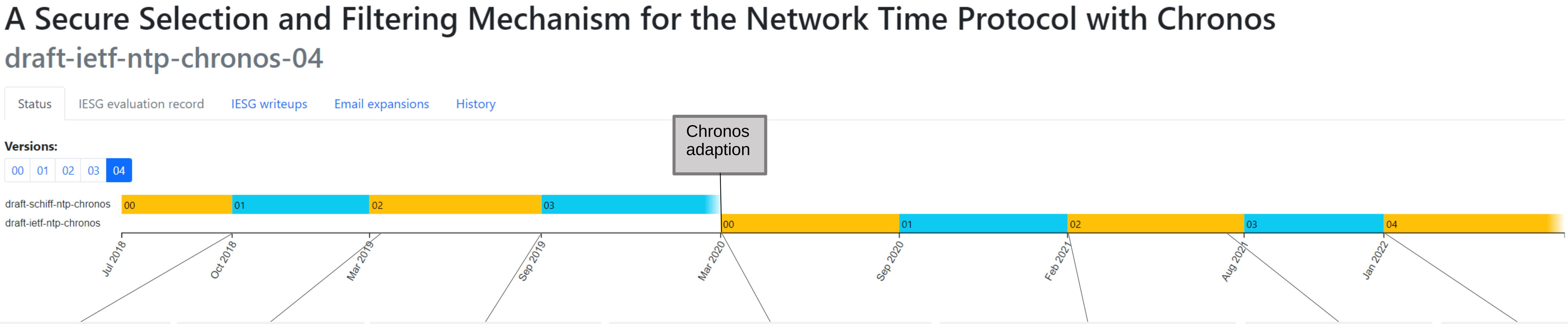
Oregon under slow-time-increase attack



Therefore, by using NTPv4 as a primary process and Chronos as a "watchdog”, Chronos watchdog matches NTPv4's accuracy and precision while significantly improving security against time shifting attacks.

Chronos Draft History

- Chronos was modified based on the comments we got from the WG.
- The main updates are the following:



A Secure Selection and Filtering Mechanism for the Network Time Protocol with Chronos

draft-ietf-ntp-chronos-04

Status IESG evaluation record IESG writeups Email expansions History

Versions:

00 01 02 03 04

- Hybrid approach – Chronos is passive till external attack indication.

- Evaluating Chronos precision and accuracy with and without attacks.

- Adding a security analysis.
- Suggesting parameters values.
- Updating link to the paper.

- Chronos as a watchdog.
- Describing the modification of NTP selection and cluster algorithms.
- Adding resiliency analysis against DDoS attacks.

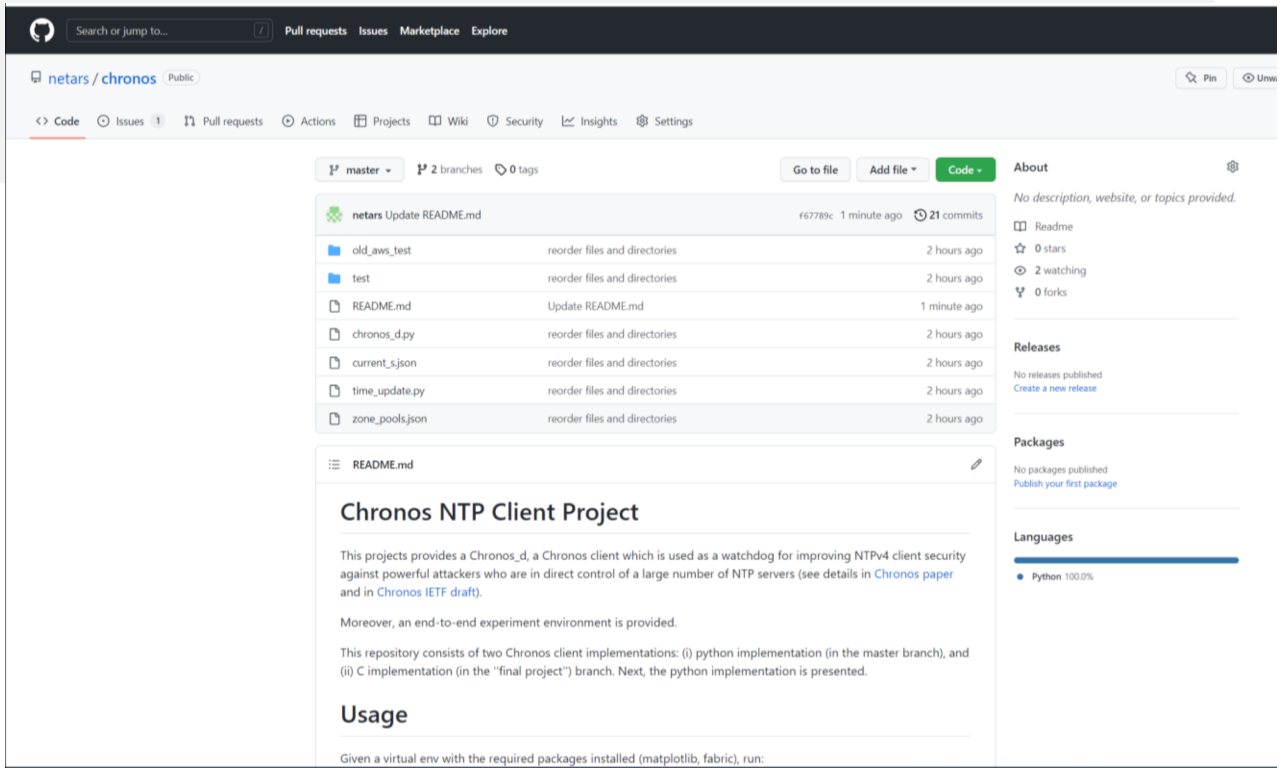
- Elaborating the watchdog mechanism
- Adding default parameters values
- Adding more details to the pseudo code

- Updating the references.
- Extending the security analysis.

- Updating draft’s title.

Open-Source Implementations

- Currently, we have two available Chronos implementations, running as a NTPv4 watchdog.
  - **Python implementation** (in the master branch) – verified
    - https://github.com/netars/chronos
  - **C implementation** (in a separate branch) – being tested
    - https://github.com/netars/chronos/tree/final_project
- Attack simulator code also available.



netars Update README.md f67789c 1 minute ago 21 commits

old_aws_test	reorder files and directories	2 hours ago
test	reorder files and directories	2 hours ago
README.md	Update README.md	1 minute ago
chronos_d.py	reorder files and directories	2 hours ago
current_s.json	reorder files and directories	2 hours ago
time_update.py	reorder files and directories	2 hours ago
zone_pools.json	reorder files and directories	2 hours ago

README.md

Chronos NTP Client Project

This projects provides a Chronos_d, a Chronos client which is used as a watchdog for improving NTPv4 client security against powerful attackers who are in direct control of a large number of NTP servers (see details in Chronos paper and in Chronos IETF draft).

Moreover, an end-to-end experiment environment is provided.

This repository consists of two Chronos client implementations: (i) python implementation (in the master branch), and (ii) C implementation (in the "final project") branch. Next, the python implementation is presented.

Usage

Given a virtual env with the required packages installed (matplotlib, fabric), run:

About

No description, website, or topics provided.

Readme
0 stars
2 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

● Python 100.0%

Group Contributors

We thank all the group contributors for the fruitful discussion:

Karen O’Donoghue

Dieter Sibold

Greg Dowd

Watson Ladd

Ulrich Windl

Erik Kline

Harlan Stenn

Danny Mayer

Miroslav Lichvar

Daniel Franke

Kristof Teichel

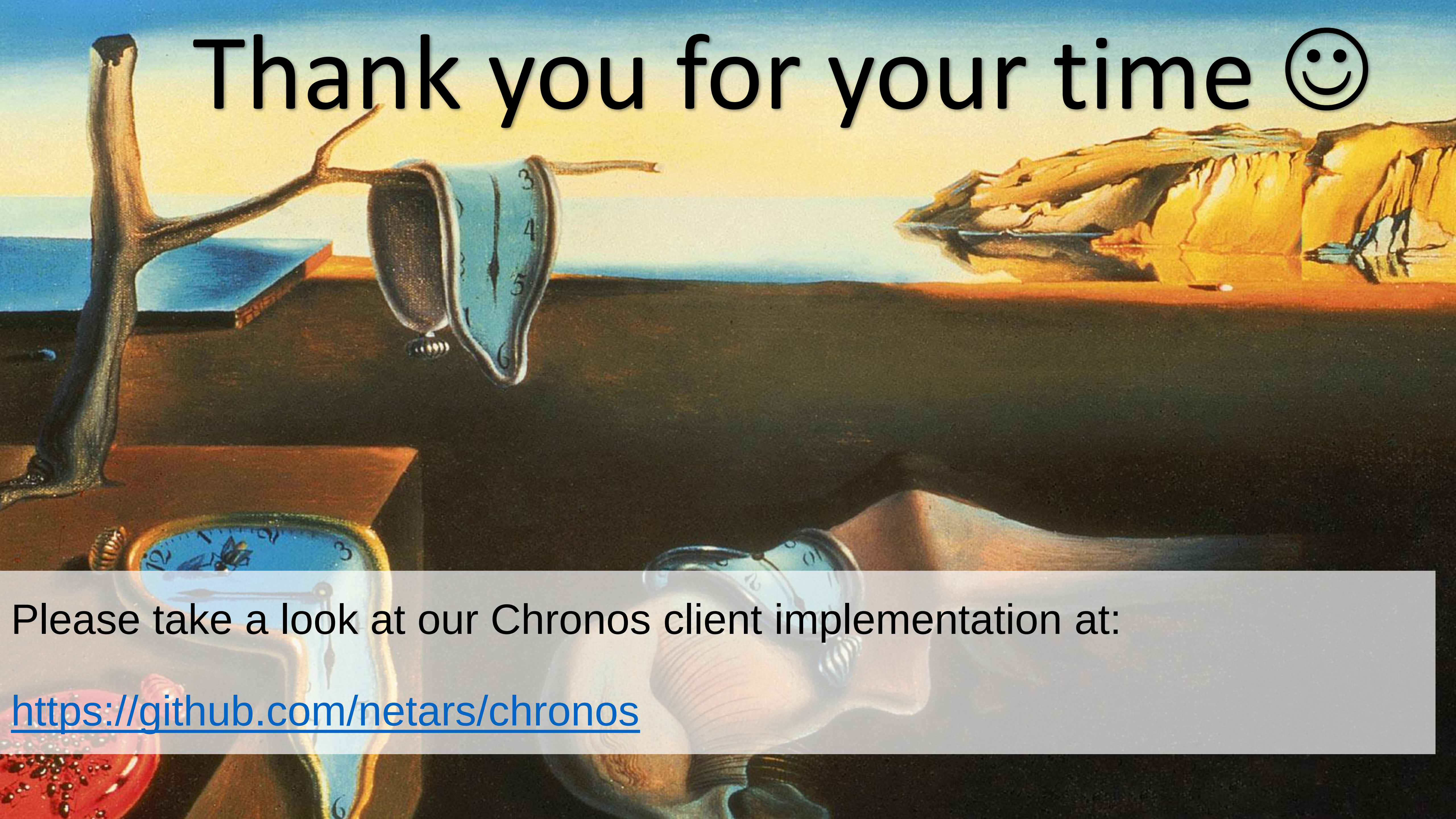
Marcus Dansarie

Yaakov. J. Stein

What’s Next?

- We believe that Chronos is ready for publication as an informational document.
  - We answered all the WG comments
  - We developed two Chronos client implementations (which are avaliable)
- We aspire to making Chronos an official watchdog for NTPv5 and are looking forward to continued collaboration with the WG.

Thank you for your time ☺



Please take a look at our Chronos client implementation at:

https://github.com/netars/chronos