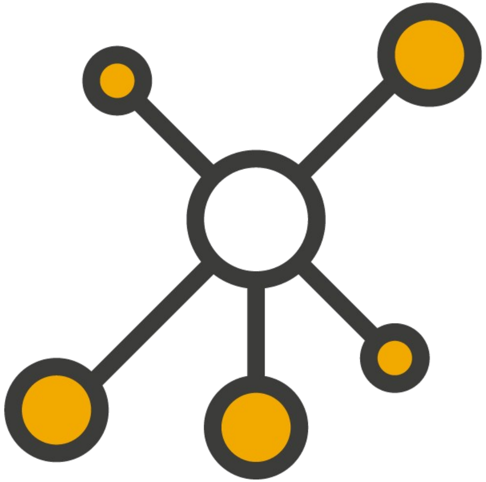
Newsgrouper, a Tcl-powered web interface to Usenet



Colin Macleod

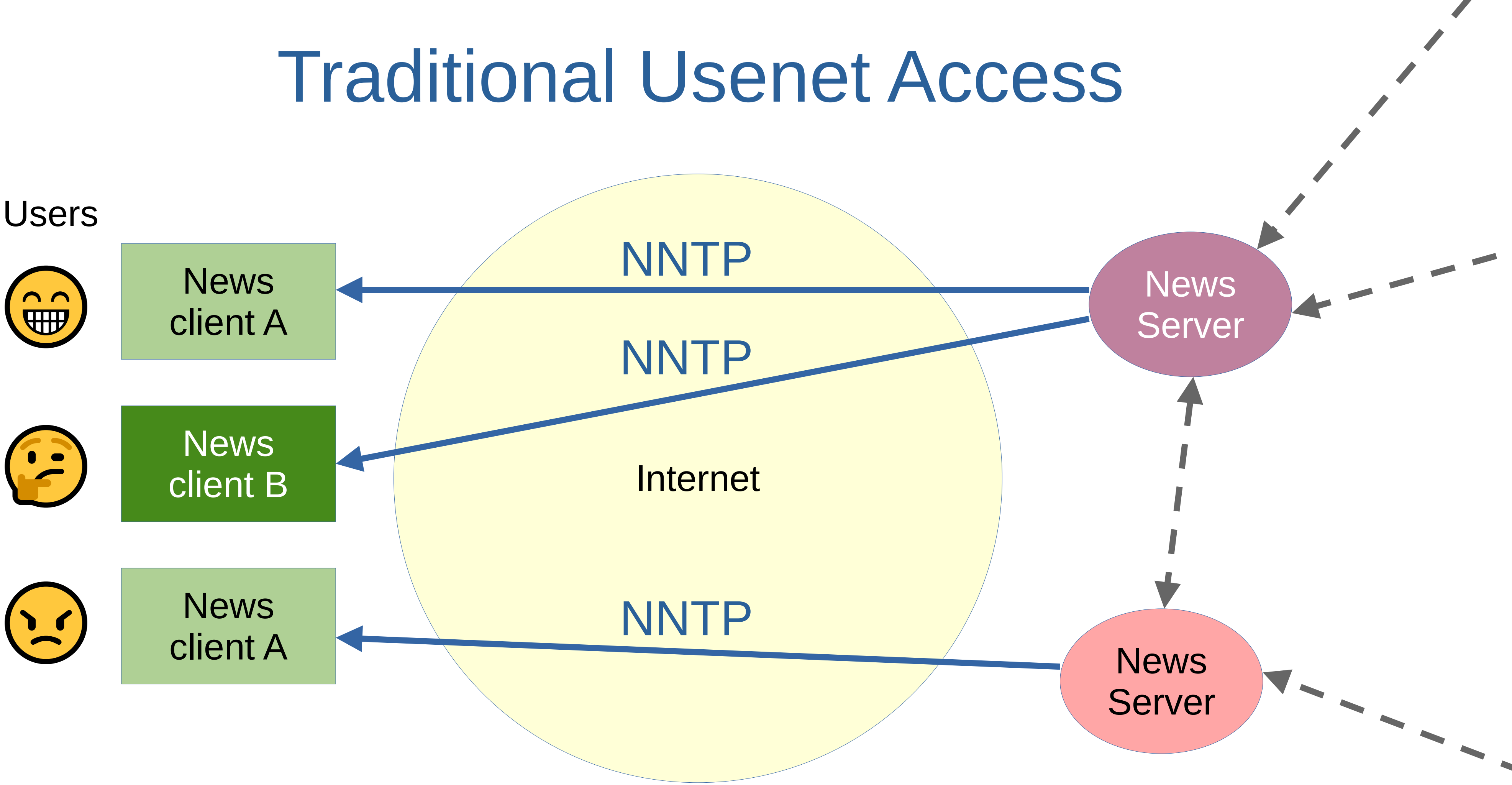
https://cmacleod.me.uk/

CGM on Tcl wiki and chat

Introduction - Usenet

- Usenet is a distributed discussion system which has been in operation since 1980, and so could claim to be the original "social network".
- Usenet is also known as *Network News*, since it was originally conceived as an electronic notice board where announcements could be posted and seen by interested people.
- It then grew to become a medium for discussion on many subjects, but due to this history the different areas are still referred to as “news” groups.
- Since the early 90s discussion of our favourite language has taken place in the group comp.lang.tcl.

Traditional Usenet Access



Access to Usenet

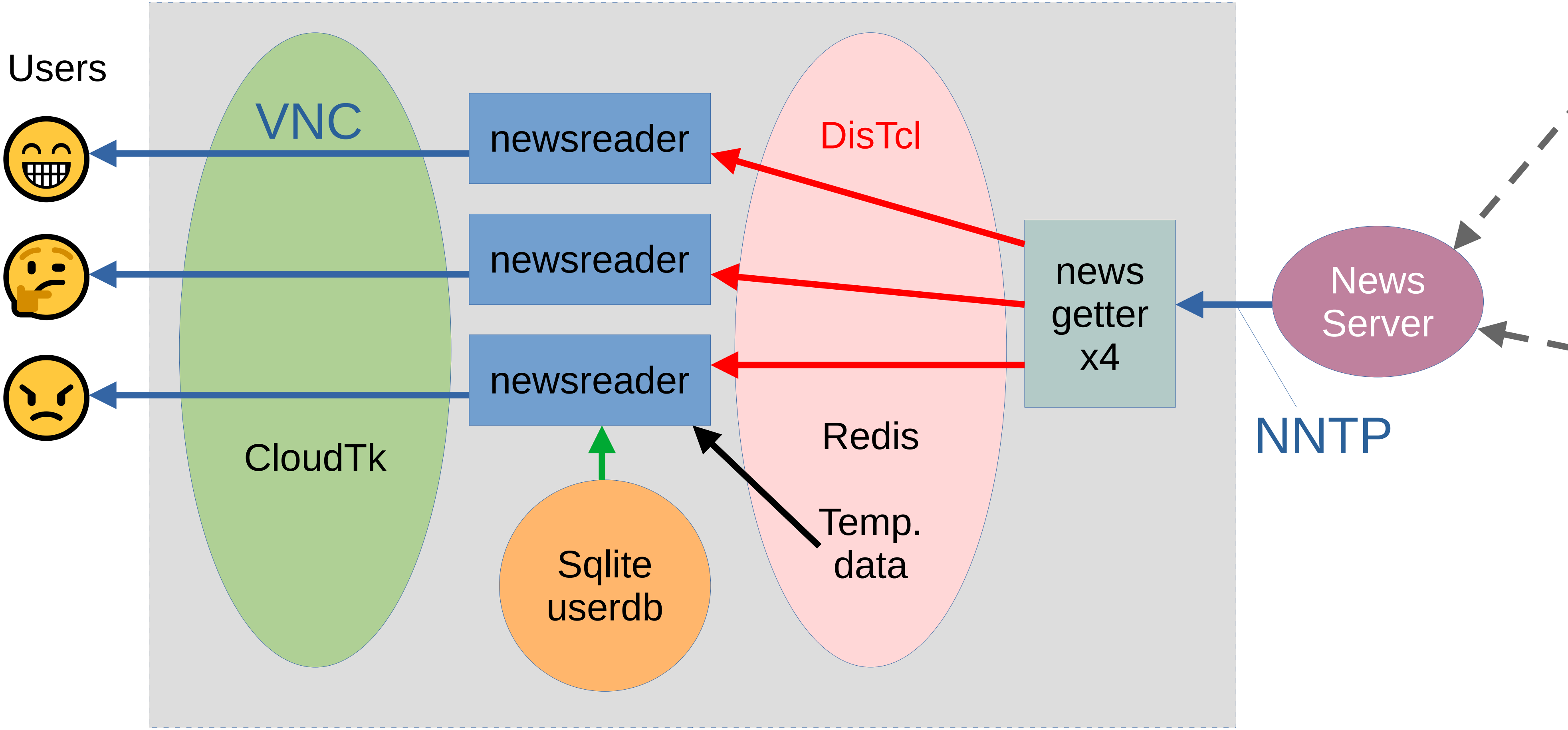
- Usenet news servers are accessed with a dedicated network protocol (NNTP) and so require client software which supports this – similar to the original situation with email.
- This need to install and configure special software now tends to exclude casual and non-technical users.
- It may not be possible in some environments, e.g. behind corporate firewalls or on mobile devices.
- Google used to provide web access through their “Groups” but this was overwhelmed with spam in 2023 and then turned off entirely in 2024.

Tcl code and Usenet

There are many existing Tcl libraries which can be used to collect, process and present Usenet data:

- **tclhttpd** - flexible web server, easily integrated with other code.
- **nntp** - the protocol needed to communicate with Usenet servers.
- **mime** - Usenet posts are closely related to emails, so the mime library can parse them.
- **sqlite** – database to store long-lived data.
- **retcl** – use Redis to store short-lived data.
- **distcl** - my distributed processing library.

Newsgrouper Phase 1 - CloudTk



DisTcl Recap

DisTcl is a simple distributed processing library which I presented at last year’s conference. Main points:

- It uses Redis as a medium for communication between client and server processes.
- It can operate across multiple machines but for Newsgrouper everything is on one machine.
- It incorporates caching of results which are stable enough to be worth caching.

Usenet Client Protocol

Usenet servers provide access using NNTP – Network News Transfer Protocol, which includes requests to:

1. Get the list of groups available
2. Get the range of article numbers available in one group
3. “Overview” – get metadata for a range of articles in one group
4. Get the text of a specific article

DisTcl and Caching

- Some of this information is fairly static and can so be cached for e.g. a week.
- But some is dynamic and so can only be cached for a few minutes.
- DisTcl uses Redis facilities to do this caching with suitable expiry times.

DisTcl and Parallelism

- The Usenet servers I am connecting to permit up to 4 connections at a time from a client system.
- So I run 4 instances of **newsgetter**, each of which opens an NNTP connection to the Usenet server.
- These all provide the same DisTcl service, so each request for that service will be handled by the first newsgetter process which is free.
- Multiple DisTcl clients can make simultaneous requests for Usenet data, these will be queued and then processed by the newsgetter servers.
- Also any one client can make multiple requests in parallel and if servers are free they will process them in parallel.

Newsgrouper UI

Since my web development knowledge is very limited, I started by implementing a user interface in Tcl/Tk, using CloudTk to make this accessible over the web. But:

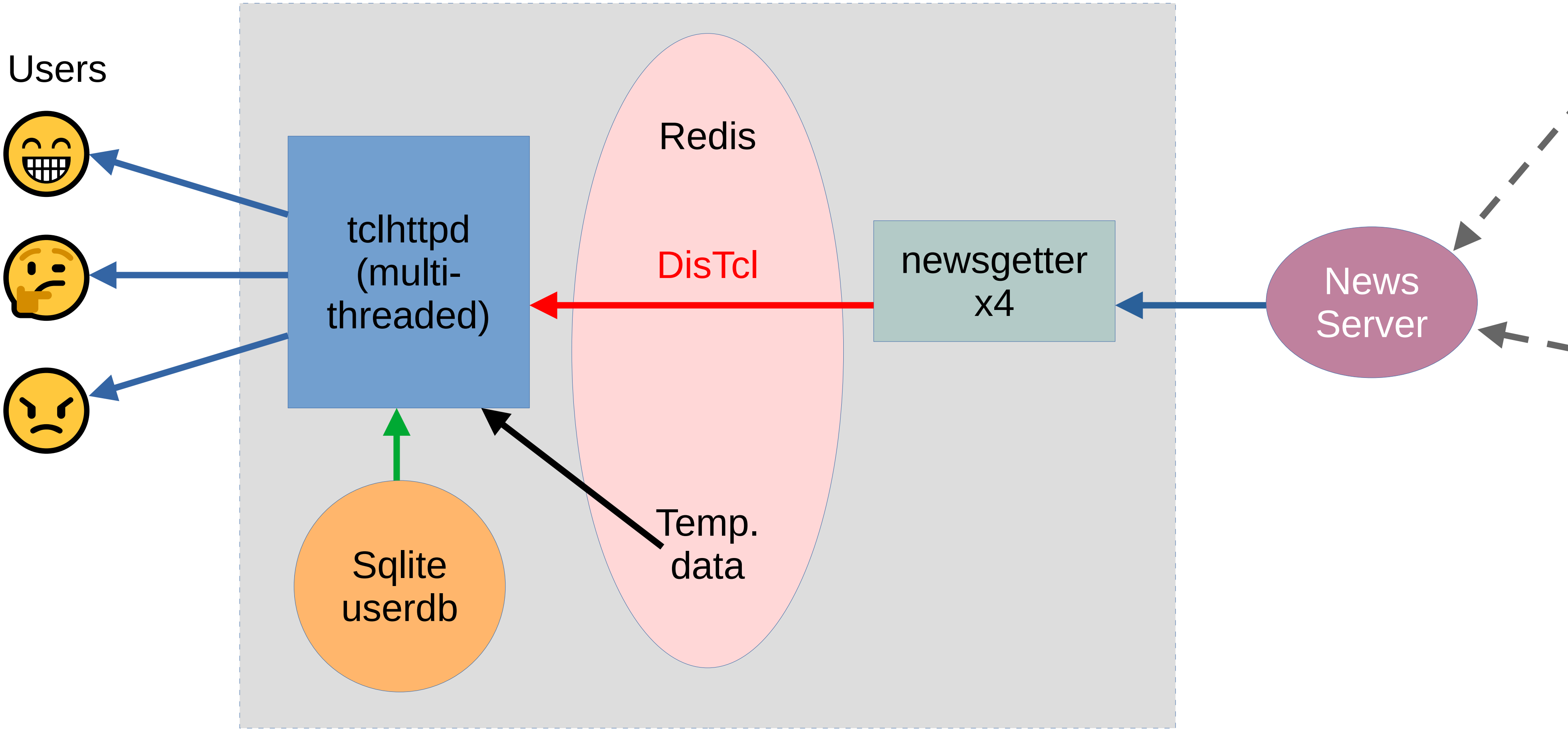
- I had great difficulty to make this properly usable on mobile devices
- I got a complaint that it was not usable by people who block javascript
- I was concerned that it would not scale up if I ever got a large number of users

Newsgrouper UI - 2

So I then switched to a more conventional web approach, generating html for each page the user sees, and aiming to make the site:

- usable on desktop and mobile platforms,
- usable even with javascript disabled,
- possible to drive from the keyboard on devices with a keyboard - this does require javascript though.

Newsgrouper Phase 2 - Web

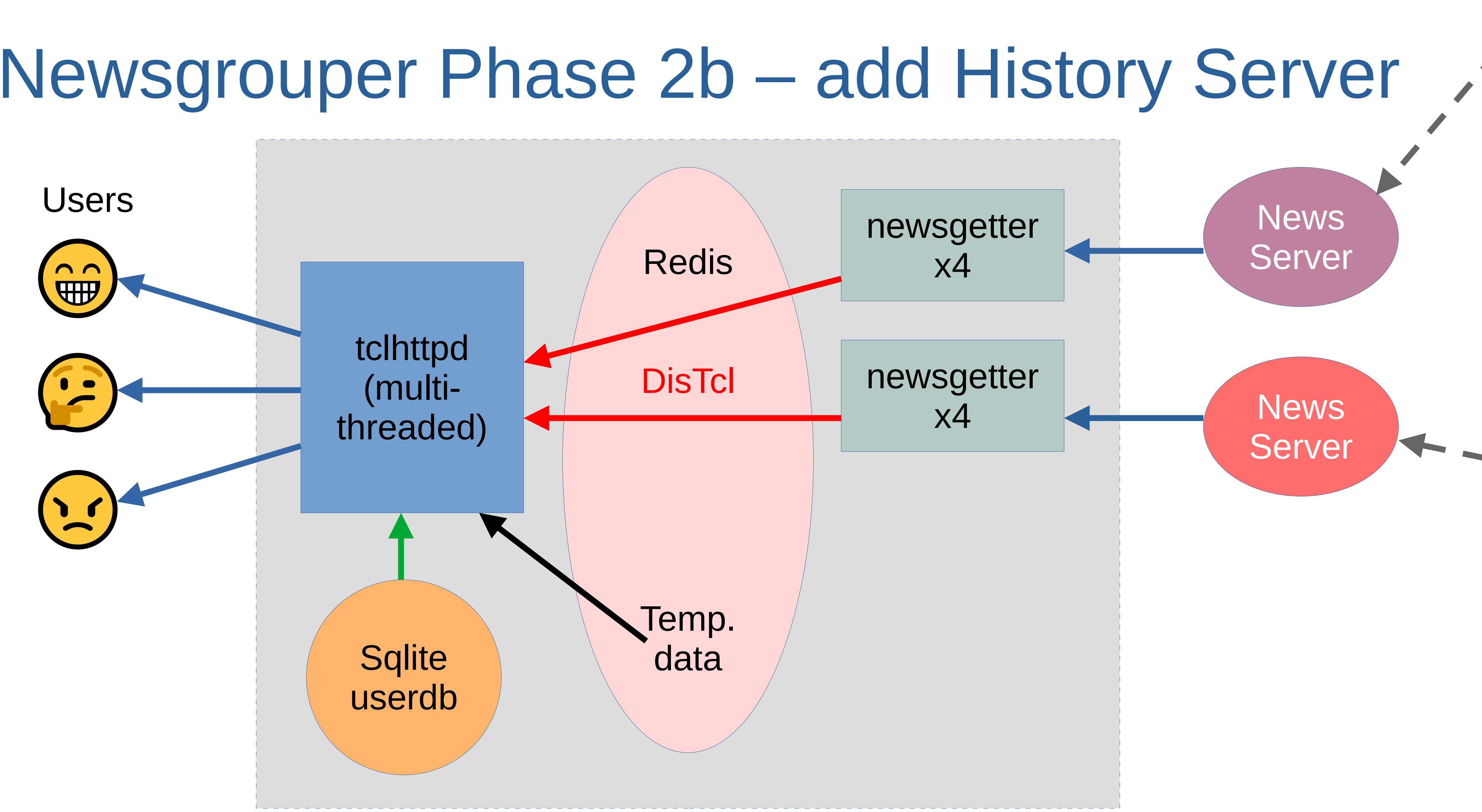


News History

Once I had access to current posting working, I started looking at access to historical content.

- Unfortunately the Usenet server I was using (eternal-september.org) then had a crash which wiped out all its history.
- So I then set up an connection to a different server (blueworldhosting.com) which had 20 years of postings online.
- I added a historical search section to the website to run searches against this history.

Newsgrouper Phase 2b – add History Server

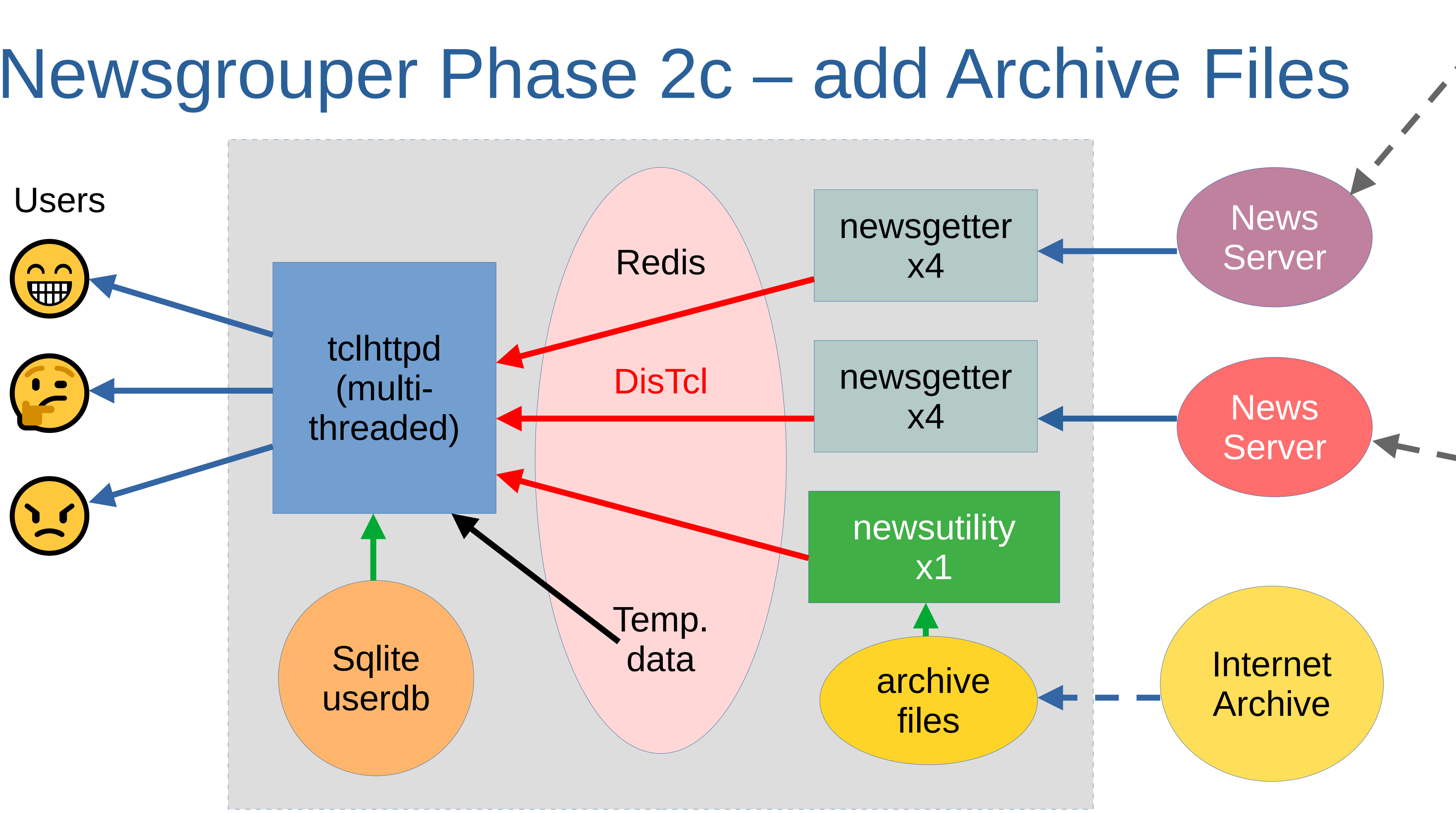


News History - 2

Next I discovered that the Internet Archive has archive files of Usenet postings from 1987 to 2013.

- These are in zipped mbox format, so not easily accessible.
- So I downloaded them to my machine (which took a few weeks) and set up a system to unzip and search them on-the-fly when requested.
- To access this I added another section to the historical search page.

Newsgrouper Phase 2c – add Archive Files

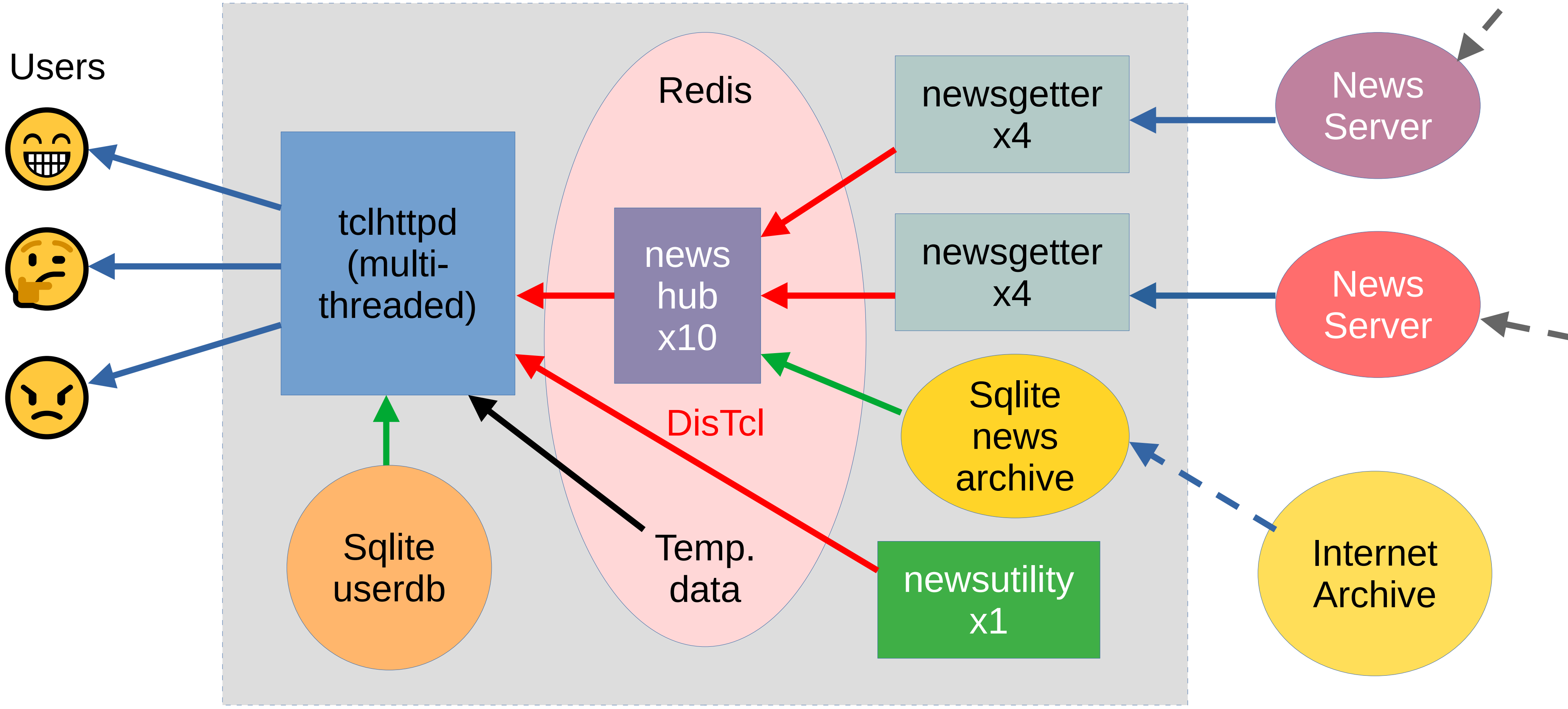


News History - 3

This is the present situation, but it’s not very satisfactory.

- The User eXperience is fragmented, having to use three different methods to find postings depending on their age.
- There are performance problems – sometimes the archive search method fails or times-out and the user gets an error page.
- Getting new postings is entirely dependent on a single Usenet server – eternal-september is good but occasionally has hiccups.

Newsgrouper Phase 3 – add News Hub



Work In Progress

- The current version has three different sources of data: nntp-current, nntp-historical, archive files. These are presented in different ways.
- I have been working on introducing a “newshub” service which will merge data from all the sources available and present it to tclhttpd as a single unified source.
- The user can then read and search each group in one place, from its early history up to the present.
- This should also make newsgrouper less dependent on a single upstream usenet provider.

Newsgrouper Demonstration

- The site is https://newsgrouper.org/
- You can try it yourself at any time.
- Unless you are in the UK, that is -
The UK’s new Online Safety Act imposes requirements which are disproportionate and impractical for a small amateur-run site like this, so I am forced to block access from my own country 🙄
- The source code is at
https://chiselapp.com/user/cmacleod/repository/newsgrouper/home

News

Grouper

Questions?

Appendices

Stuff that got squeezed out…

What is Usenet? - 2

- Usenet now comprises many thousands of groups for different subjects, including comp.lang.tcl for discussion of The Cool Language.
- In some respects Usenet is primitive. Natively it supports only text content. Later various workarounds were established to enable binary files to be distributed, but this project supports text content only.
- Usenet operates as a collaborative effort between those servers which support it. There is no single company or institution in control. There are no adverts and no algorithmic promotion of particular content.

Usenet User Interfaces

- Usenet has a lot in common with email, except that Usenet is designed for public communication while email is for communication between individuals.
- Both predate the world-wide web, and both were originally used only through application-specific client software.
- Now web interfaces to email are common, but web interfaces to Usenet are rare.
- The need to install and configure dedicated software is a barrier to discovery and casual exploration of Usenet.

Usenet User Interfaces - 2

- Google used to support web access to Usenet through their Groups system, but they disconnected that from Usenet in February 2024.
- Newsgrouper is one attempt to fill the resulting gap and maintain easy access to comp.lang.tcl and many other Usenet communities.
- A few other web interfaces to Usenet exist, but as far as I am aware Newsgrouper is the only one which supports the full range of groups for reading, posting and searching.

Newsgrouper UI State

Switching from Tk to html completely changes how you manage state for each user:

- with CloudTk each user has a separate process on the server which encapsulates their state.
- with a webserver generating html, there is no such continuity between the code handling one user request and the next. So you need to store user state between requests and retrieve it on the next request for the same user.

Newsgrouper UI State - 2

Newsgrouper manages user state by:

- Setting a single “cookie” to identify the user.
- Storing short-lived state data within tclhttpd in *thread shared variables* by using the tsv::* commands in Tcl’s Thread extension.
- Caching longer-lived state data in Redis by using the Retcl extension.
- Both of these stores use the user id as part of their key to identify the state for each user.

Newsgrouper Architecture - 2

Newsgrouper's functionality is divided among a group of processes which communicate through DisTcl. The components are:

- tclhttpd - the web server. This generates web pages in response to user requests. It is a DisTcl client.
- newsgetter – DisTcl servers which use NNTP to get index data and postings from Usenet servers.
- newsutility – A DisTcl server for various other functions including running searches on downloaded historical Usenet archive files.