Presentation at the 89th IETF Meeting

# On the Applicability of Reliable Server Pooling for Virtualised Network Function Resource Pooling

**Thomas Dreibholz, dreibh@simula.no**

Simula Research Laboratory

**4 March 2014**

# Contents

- What is required for VNFPOOL?
- What is provided by Reliable Server Pooling?
- Discussion!

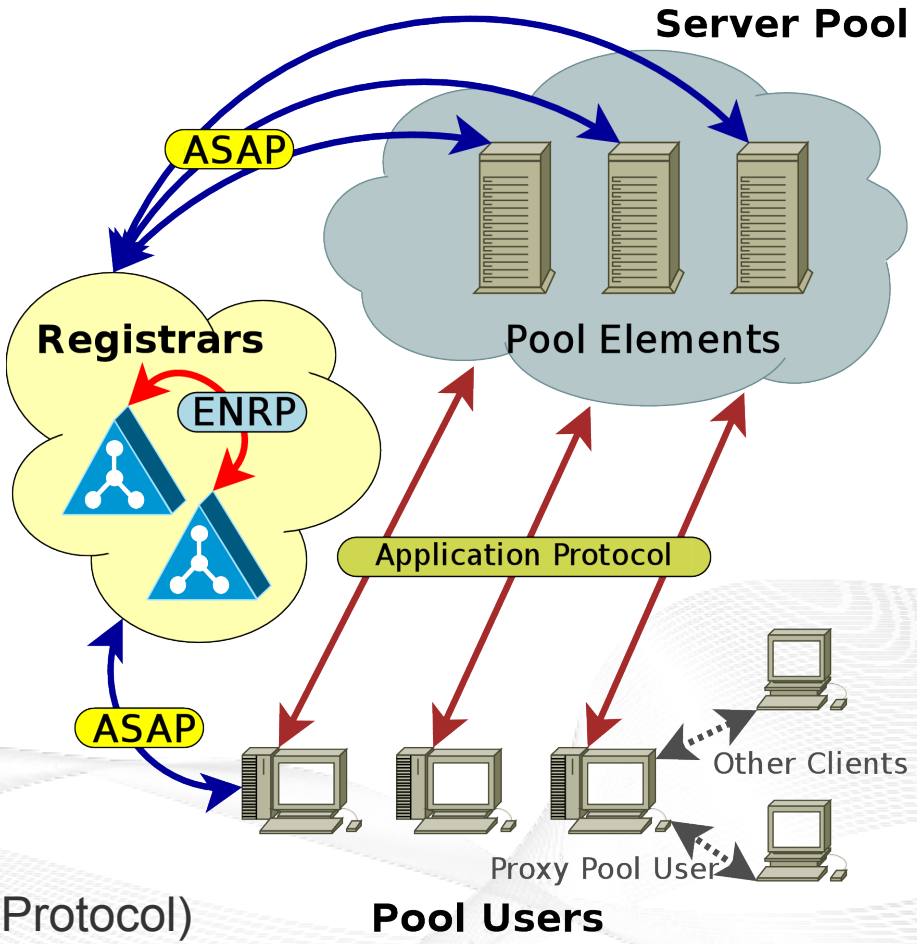
# What is required for VNFPOOL?

- **Virtualised Network Function (VNF)**:
  - provides the same function as the equivalent network function
  - Examples: firewall, load balancer, etc.
  - A VNF is a single point of failure => **redundancy is required**!
- **Virtualised Network Function Resource Pooling (VNFPOOL)**
  - VNF pool = group of VNF instances providing same function
  - Pool Manager (PM):
    - Management of VNF instances
    - Instance selection, monitoring, …
  - Many similarities with **Reliable Server Pooling (RSerPool)**!

**Can we avoid to reinvent the wheel again?**

# What is provided by Reliable Server Pooling?

- Terminology:
  - **Pool Element (PE)**: Server
  - Pool: Set of PEs
  - PE ID: ID of a PE in a pool
  - Pool Handle: Unique pool ID
  - Handlespace: Set of pools
  - Pool **Registrar (PR)**
  - **Pool User (PU)**: Client




- Protocols:
  - **ASAP** (Aggregate Server Access Protocol)
  - **ENRP** (Endpoint Handlespace Redundancy Protocol)

- RFCs: 5351–5356, 5525 + a couple of I-Ds

# Pool Element Selection and Failover

- **Pool Element selection**
  - Based on pool member selection policies (pool policy)
  - Pool policies:
    - Least Used, Round Robin, Random, ...
    - Priority, …
    - Easy to add new pool policies
- **Failover support**
  - Application-specific, but RSerPool can help the application
  - Cookies: client-based state sharing
  - Business Card:
    1) PE tells PU the list of PEs to make a failover to (“last will”)
    2) Symmetric case: PU is a PE in another pool

**This is the theory, but what about “running code”?**

# *RSPLIB* – The Reference Implementation

- Design decisions:
  - **Open Source**
  - **Platform-independent**
    - Currently: Linux, FreeBSD, MacOS X, Solaris
    - Easy portability
  - Implemented in ANSI-C
- Basic components:
  - ***RSPLIB* library for PUs and PEs**
    - ASAP protocol (PU/PE side)
  - **Registrar**
    - ASAP protocol (PR side)
    - ENRP protocol
  - Demo system and **many examples**


**See http://www.iem.uni-due.de/~dreibh/rserpool/ for details!**

# Discussion!
# RSerPool for Virtualised Network Function Resource Pooling

- **What is already provided by RSerPool?**
  - Pool management
  - PE selection
  - Session management with help of Business Cards (“last will”)/Cookies
- **What is needed in addition?**
  - (MP)TCP as additional/alternative transport protocol? → should be easy!
  - Possibly add some special pool policies?
- **Out of scope of RSerPool itself** (application-specific; to be built on top):
  - State synchronisation for VNFPOOL
  - VNFPOOL Pool Manager as an RSerPool-based service?
- Draft documents
  - **https://tools.ietf.org/html/draft-dreibholz-vnfpool-rserpool-applic-00**
  - **https://tools.ietf.org/html/draft-dreibholz-rserpool-nextgen-ideas-01**

# Any Questions?

**Thomas Dreibholz, dreibh@simula.no**


**http://www.iem.uni-due.de/~dreibh/rserpool/**

NORNET

**https://www.nntb.no**