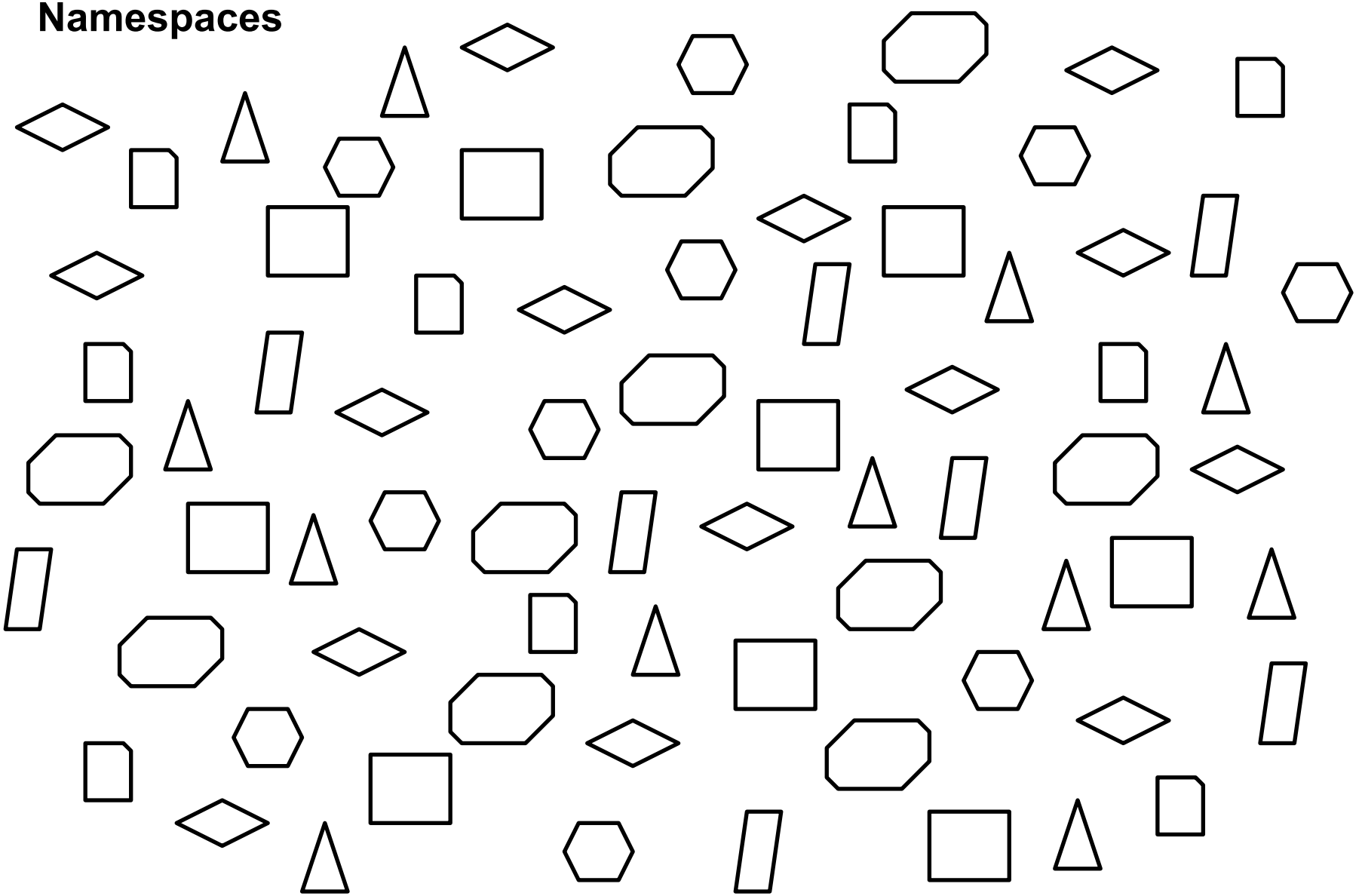
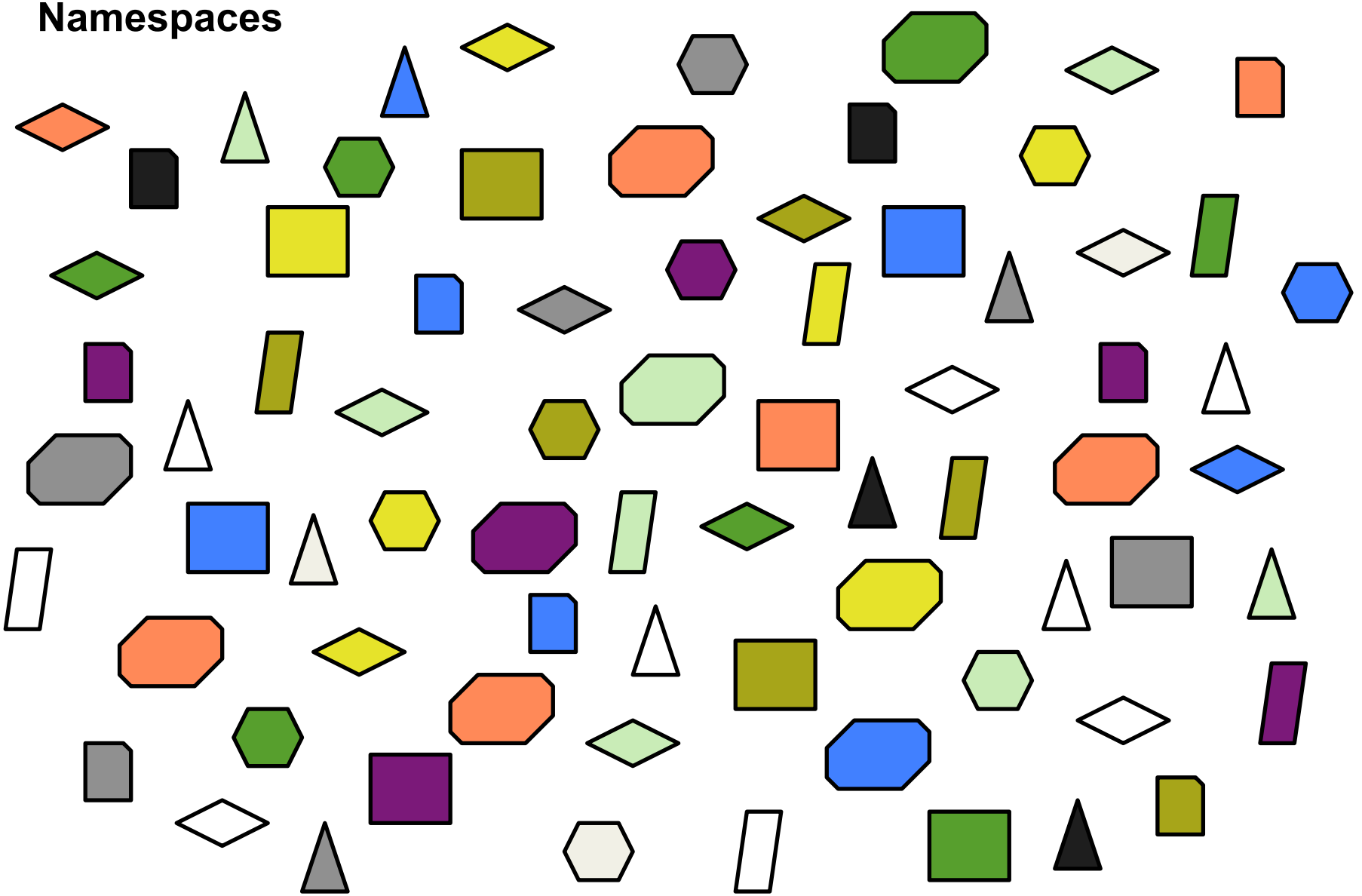
Die Technologie unter der Haube

Namespaces

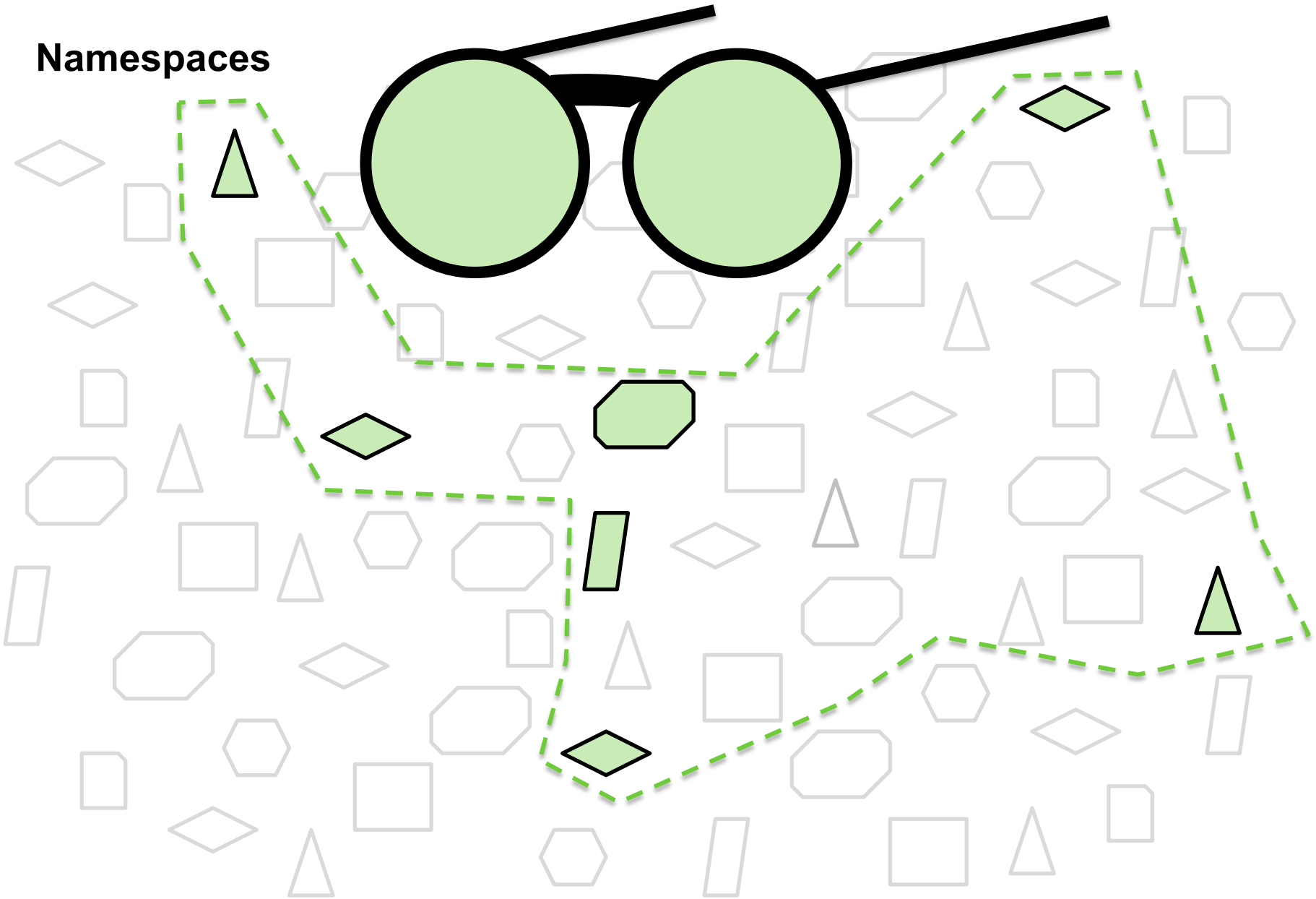


cassini GUIDING AHEAD

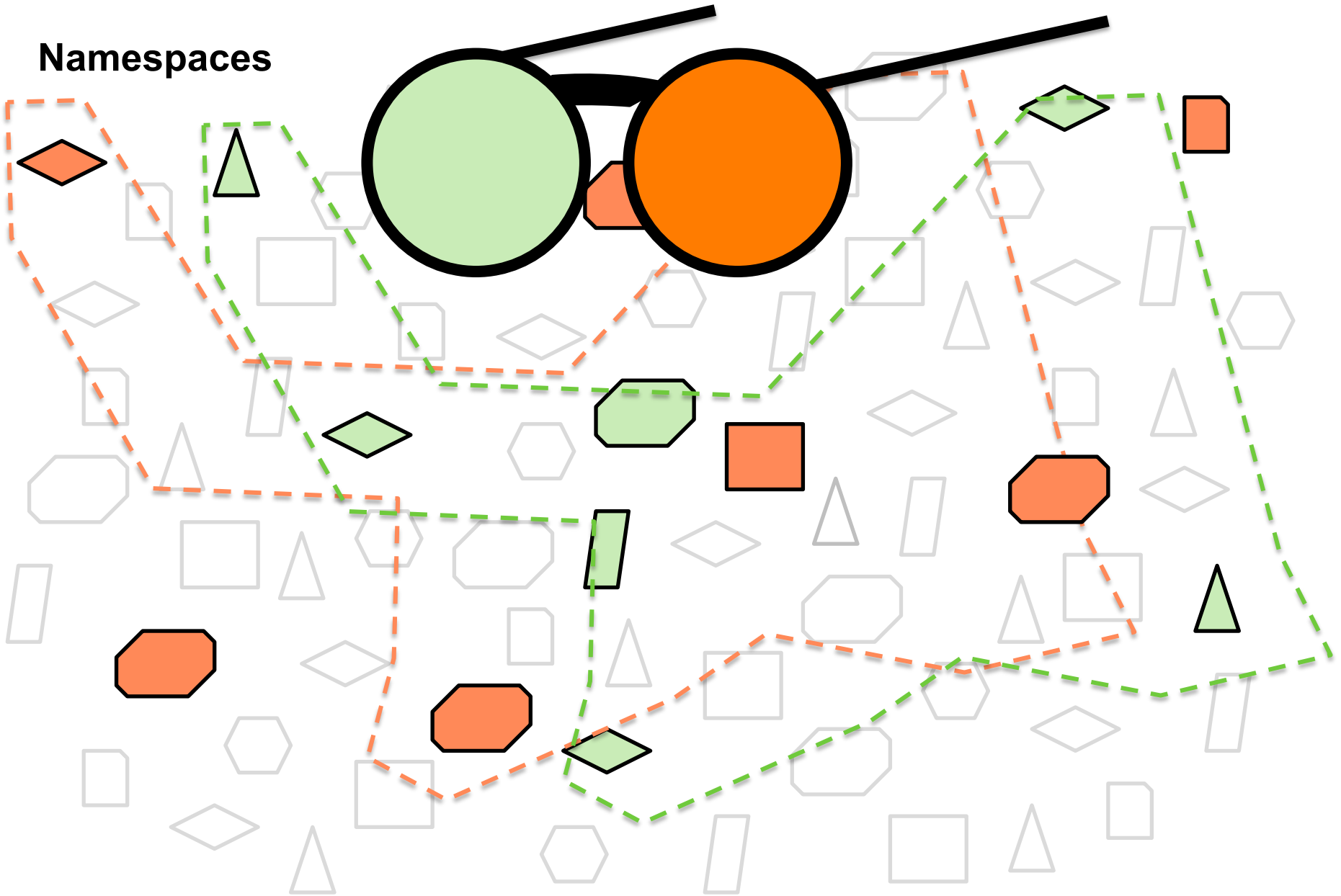
Namespaces



Namespaces



Namespaces



Namespaces (*)

Mount Namespace	„isolate the set of filesystem mount points seen by a group of processes.“
UTS Namespace	„.. allows each container to have its own hostname and NIS domain name.“
IPC Namespace	„... isolate certain interprocess communication (IPC) resources“
PID Namespace http://lwn.net/Articles/531419/ http://lwn.net/Articles/532748/	„... isolate the process ID number space.“ „... processes in different PID namespaces can have the same PID“. „... a process has two PIDs: the PID inside the namespace, and the PID outside the namespace on the host system“ „... can be nested.“
Network Namespace http://lwn.net/Articles/580893/	„ ... each network namespace has its own network devices, IP addresses, IP routing tables, /proc/net directory, port numbers, and so on“
User Namespace http://lwn.net/Articles/532593/	completed in Linux 3.8 „... a process's user and group IDs can be different inside and outside a user namespace.“

(*) http://lwn.net/Articles/531114/

Process Namespaces

- Einen Container mit Shell starten,
- die Container ID ermitteln und merken

```
~$ docker –tdi ubuntu:14.04 /bin/bash

~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
88d152af788a        ubuntu:14.04        /bin/bash           26 minutes ago
Up 26 minutes                           hungry_mcclintock
```

Process Namespaces

- Terminalfenster teilen bzw. zusätzliches öffnen.
- An den Container „attachen“ und die Prozessliste ansehen.

```
~$ docker attach 88d152

root@88d152af788a:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 15:12 ?        00:00:00 /bin/bash
root        18     1  0 15:41 ?        00:00:00 ps –ef
```

Process Namespaces

- Auf dem Host die Prozess-ID des Containers ermitteln.
- Den Prozess anzeigen lassen

```
~# docker inspect -f '{{ .State.Pid }}' 88d152
 18851

~# ps u -p 18851
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      18851  0.0  0.3  18156  2004 pts/3    Ss+  15:12   0:00 /bin/bash

~# ps faux | grep -C 5 18851
root       7093  0.8  3.3 438108 16728 ?        Ssl  14:53   0:23 /usr/bin/docker –d …
root      18851  0.0  0.3  18156  2004 pts/3    Ss+  15:12   0:00  \_ /bin/bash
```

Process Namespaces

- Auf dem Host die Namespaces aus /proc auslesen
- Mit der (erweiterten) Prozessliste vergleichen

```
~# ls -al  /proc/18851/ns/
total 0
dr-x--x--x 2 root root 0 Sep  2 15:37 .
dr-xr-xr-x 9 root root 0 Sep  2 15:12 ..
lrwxrwxrwx 1 root root 0 Sep  2 15:37 ipc -> ipc:[4026532172]
lrwxrwxrwx 1 root root 0 Sep  2 15:37 mnt -> mnt:[4026532170]
lrwxrwxrwx 1 root root 0 Sep  2 15:37 net -> net:[4026532175]
lrwxrwxrwx 1 root root 0 Sep  2 15:37 pid -> pid:[4026532173]
lrwxrwxrwx 1 root root 0 Sep  2 15:37 user -> user:[4026531837]
lrwxrwxrwx 1 root root 0 Sep  2 15:37 uts -> uts:[4026532171]

~# ps -eo ipcns,cgroup,pid,args | grep 18851
4026532172 10:perf_event:/docker/88d15 18851 /bin/bash

~# ps -eo netns,cgroup,pid,args | grep 18851
4026532175 10:perf_event:/docker/88d15 18851 /bin/bash
```