

The Next Frontier of Cloud Computing is in the Clouds, Literally

Silviu Craciunas, Andreas Haas
Christoph Kirsch, Hannes Payer
Harald Röck, Andreas Rottmann
Ana Sokolova, Rainer Trummer

Joshua Love
Raja Sengupta

Universität Salzburg

UC Berkeley



Cornell University, AI-Sys-Robo Seminar, February 2011



The JAviator

javiator.cs.uni-salzburg.at

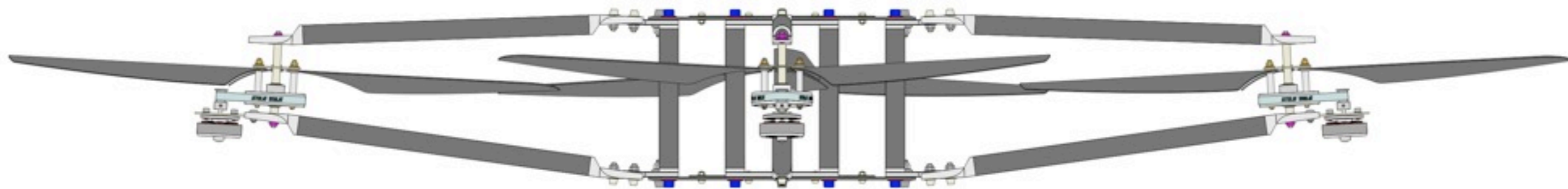
Quad-Rotor Helicopter



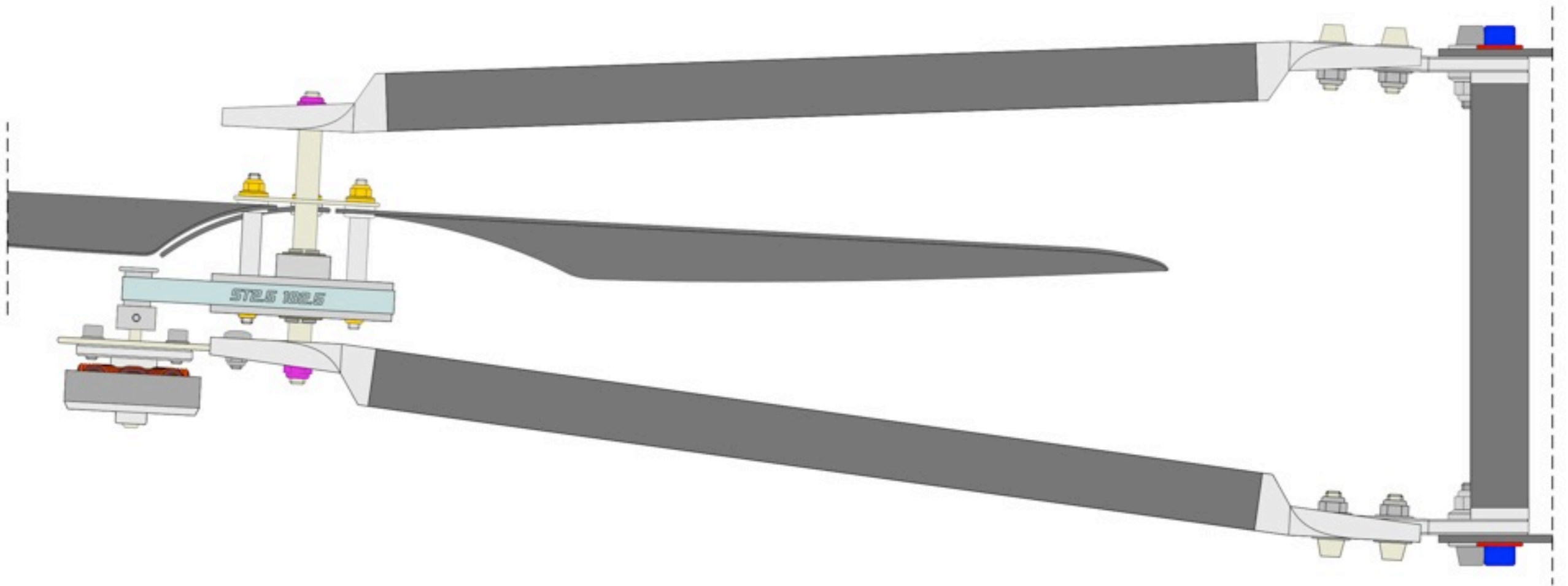
- all carbon, titanium, aluminum design
- custom motors
- 1.3m diameter
- ~2.2kg weight
- +2kg payload
- ~40min (empty)
- ~10min (full)

[AIAA GNC 2008]

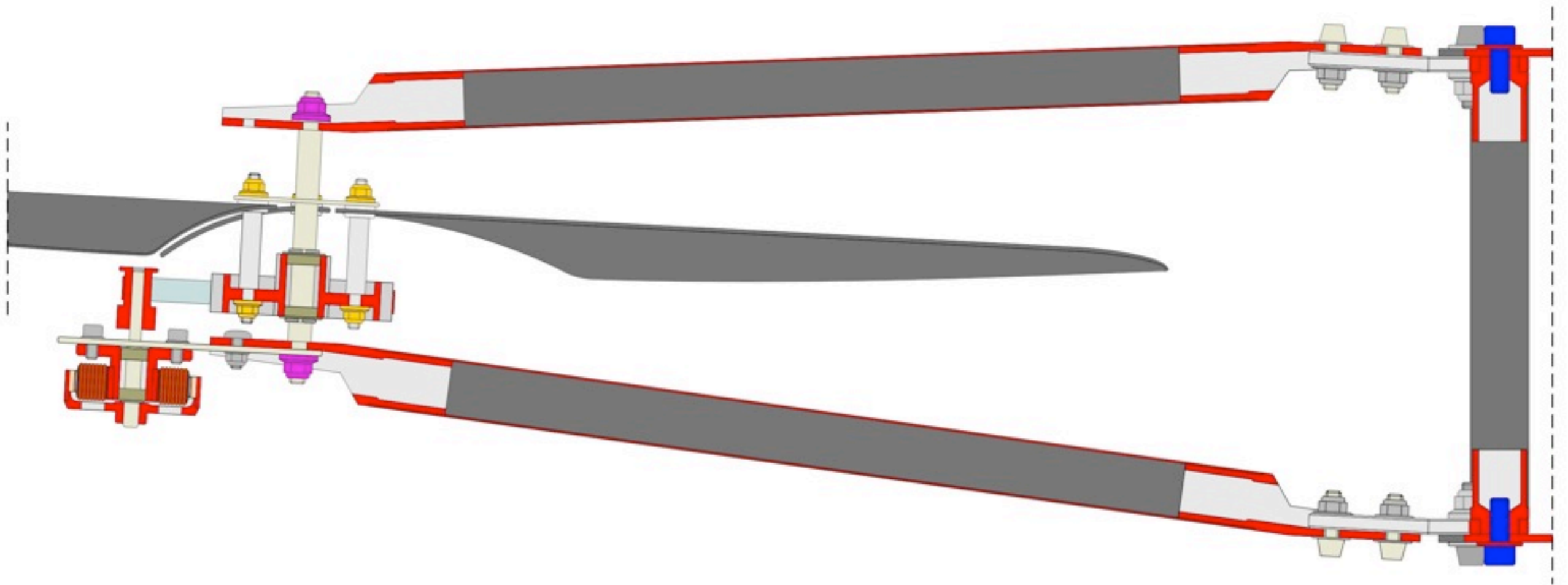
Open Source Blueprints



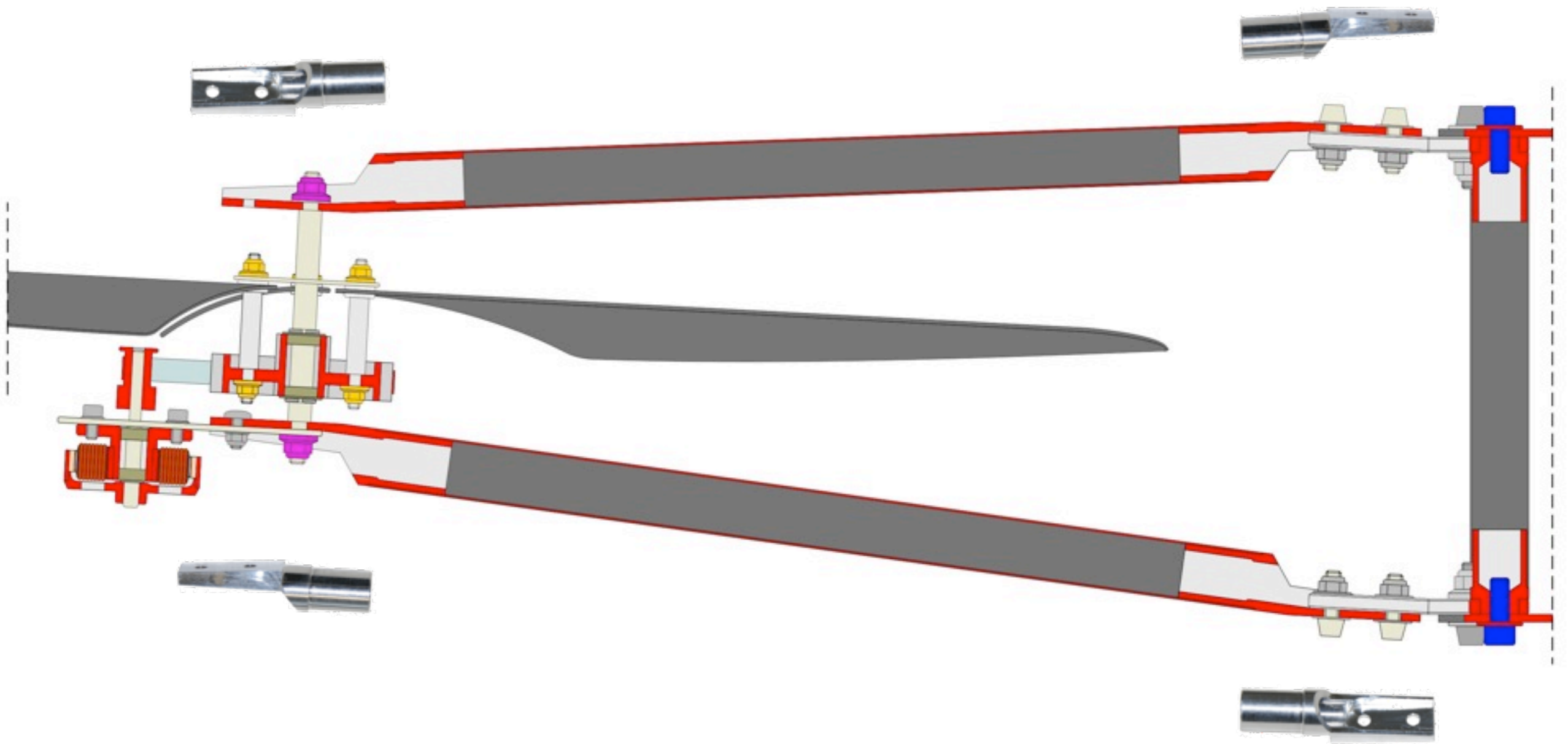
Minimal # of Different Parts



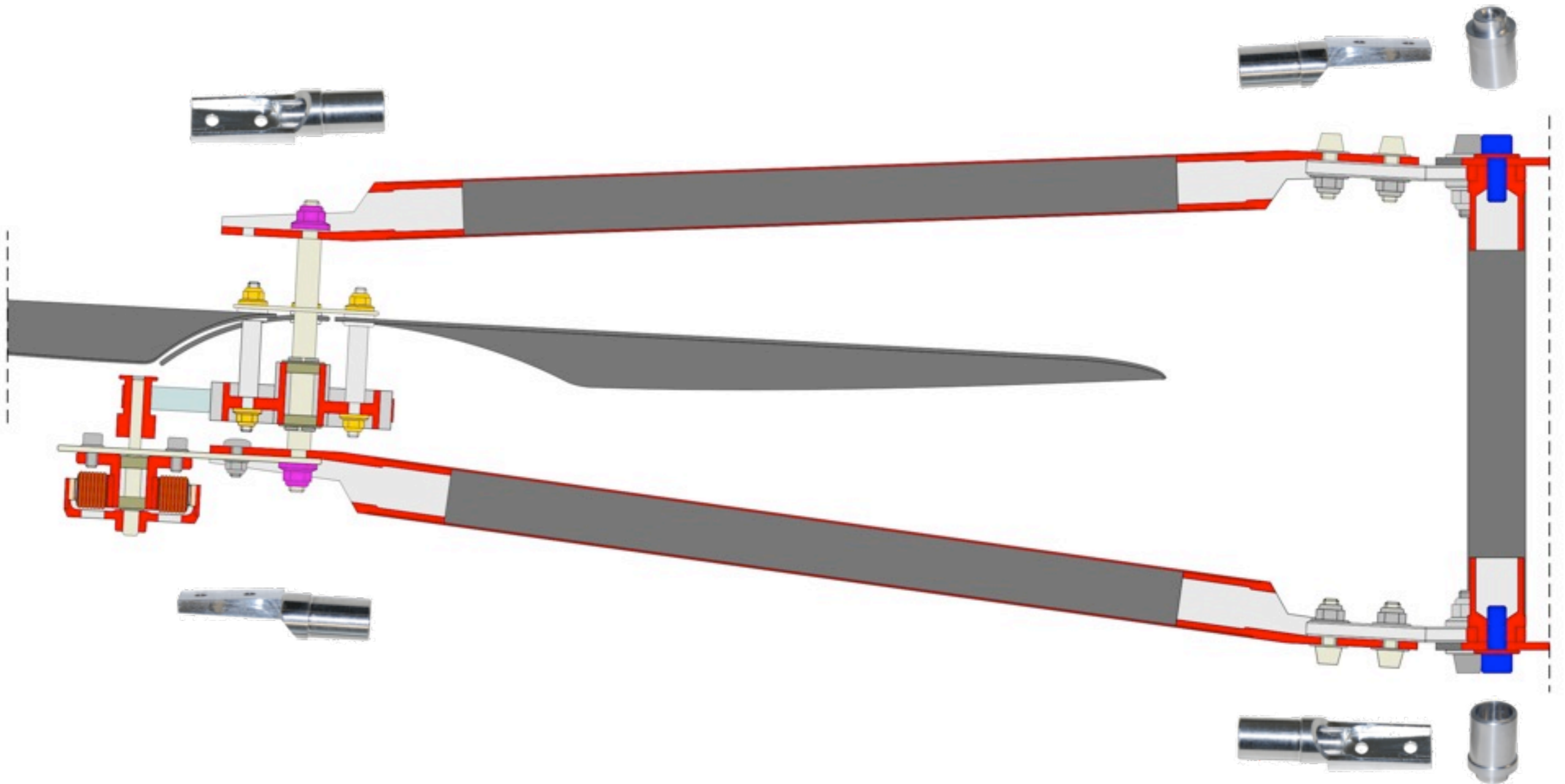
Minimal # of Different Parts



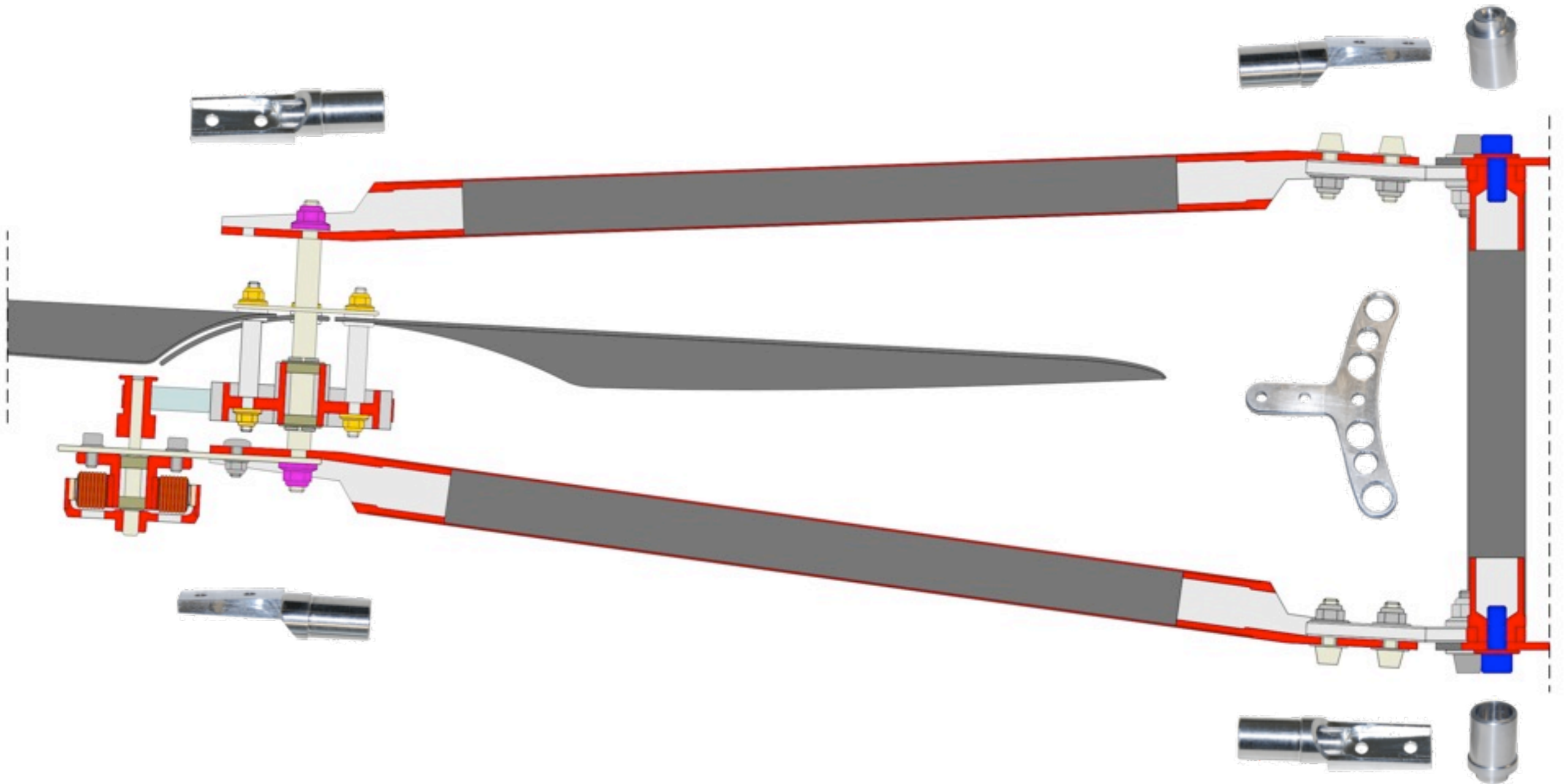
Minimal # of Different Parts



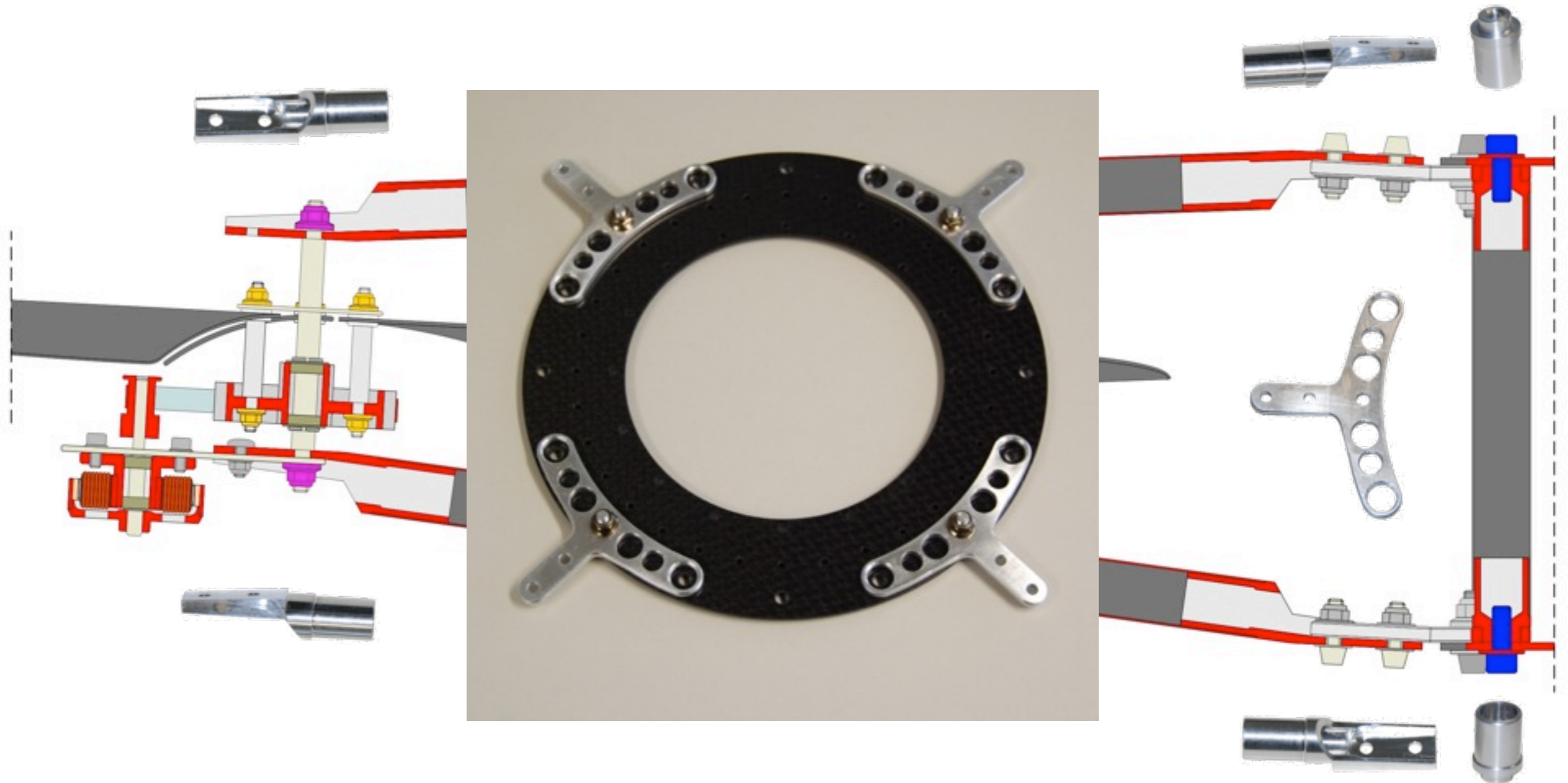
Minimal # of Different Parts



Minimal # of Different Parts



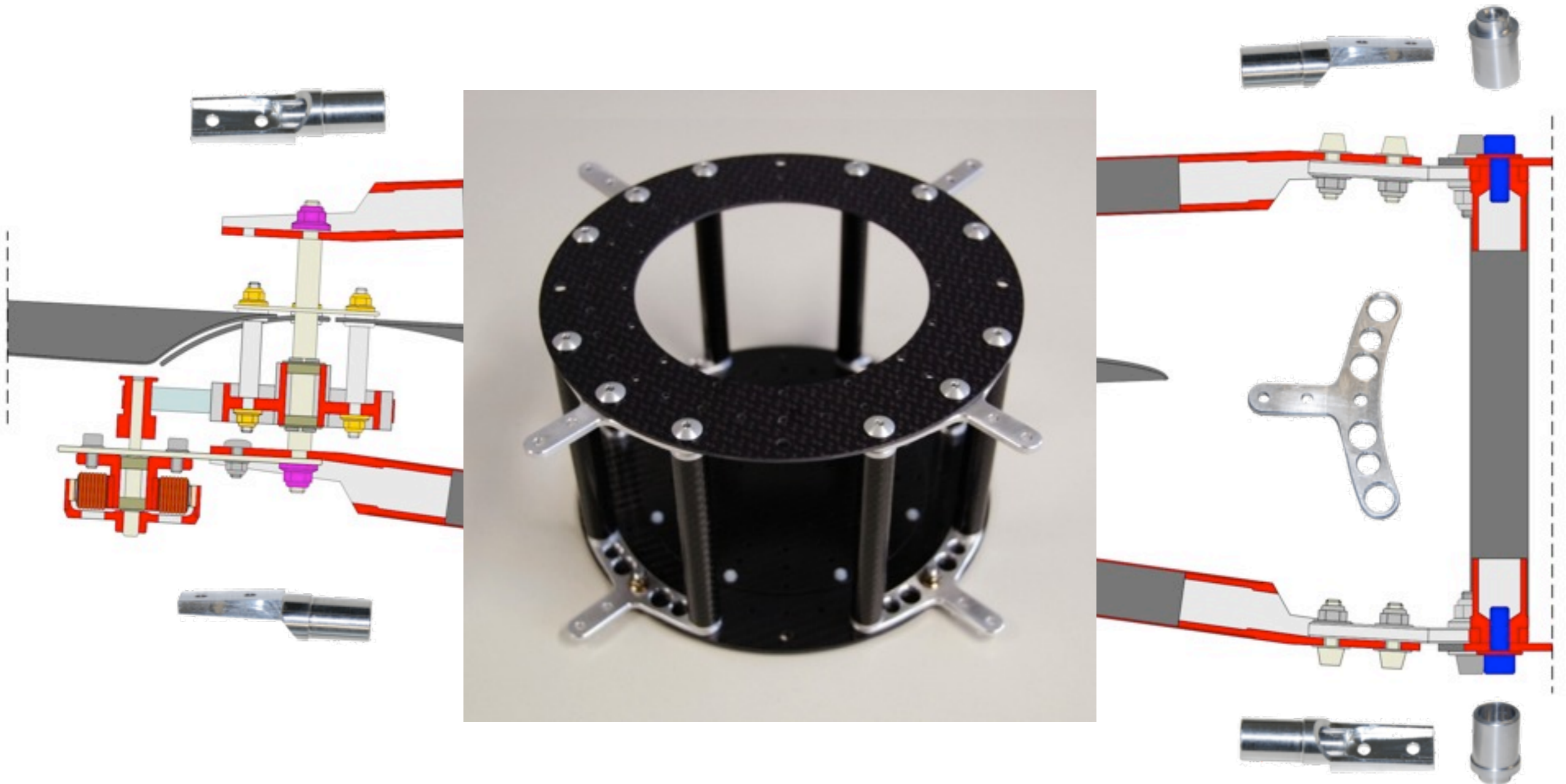
Minimal # of Different Parts



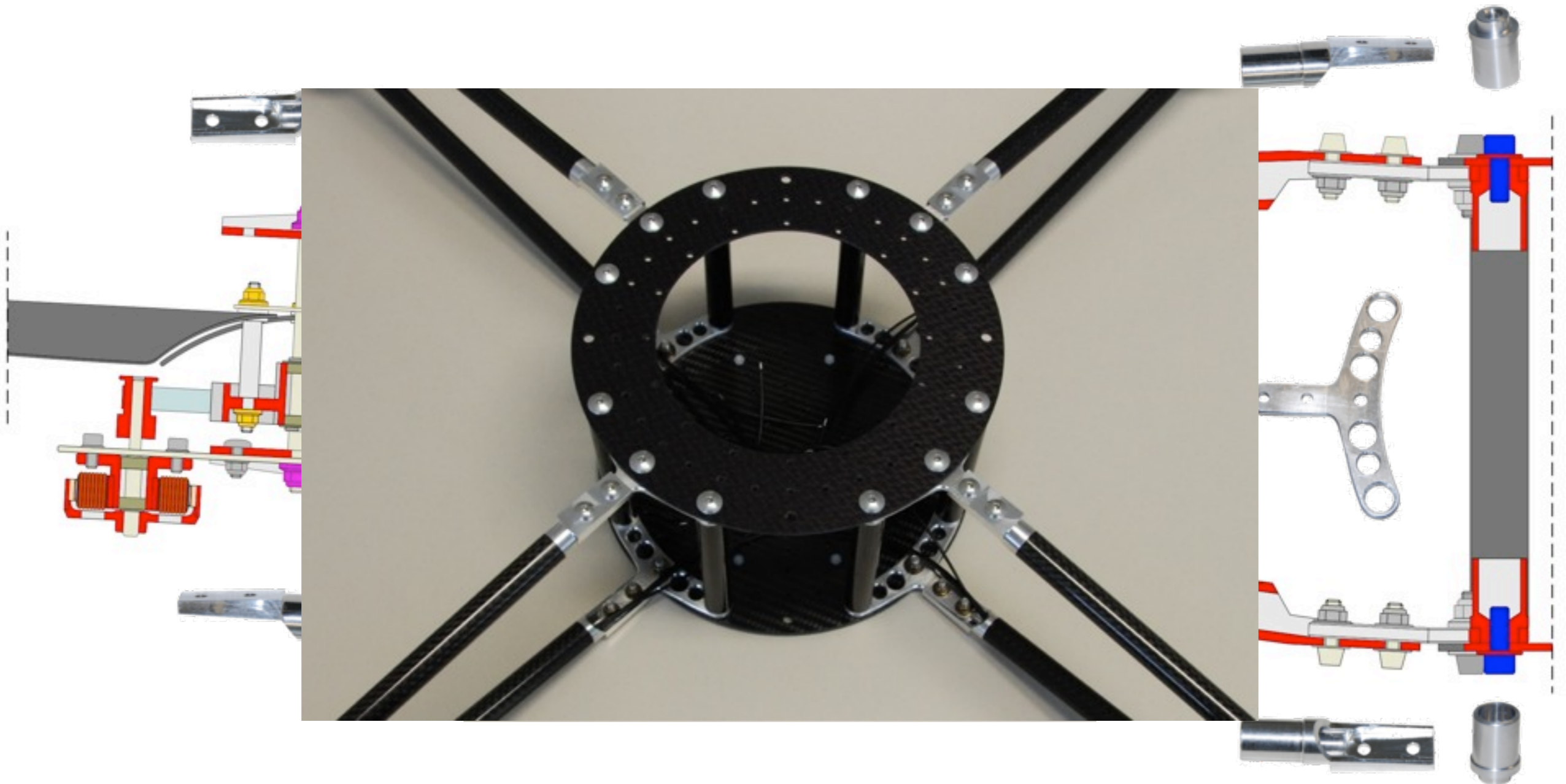
Minimal # of Different Parts



Minimal # of Different Parts



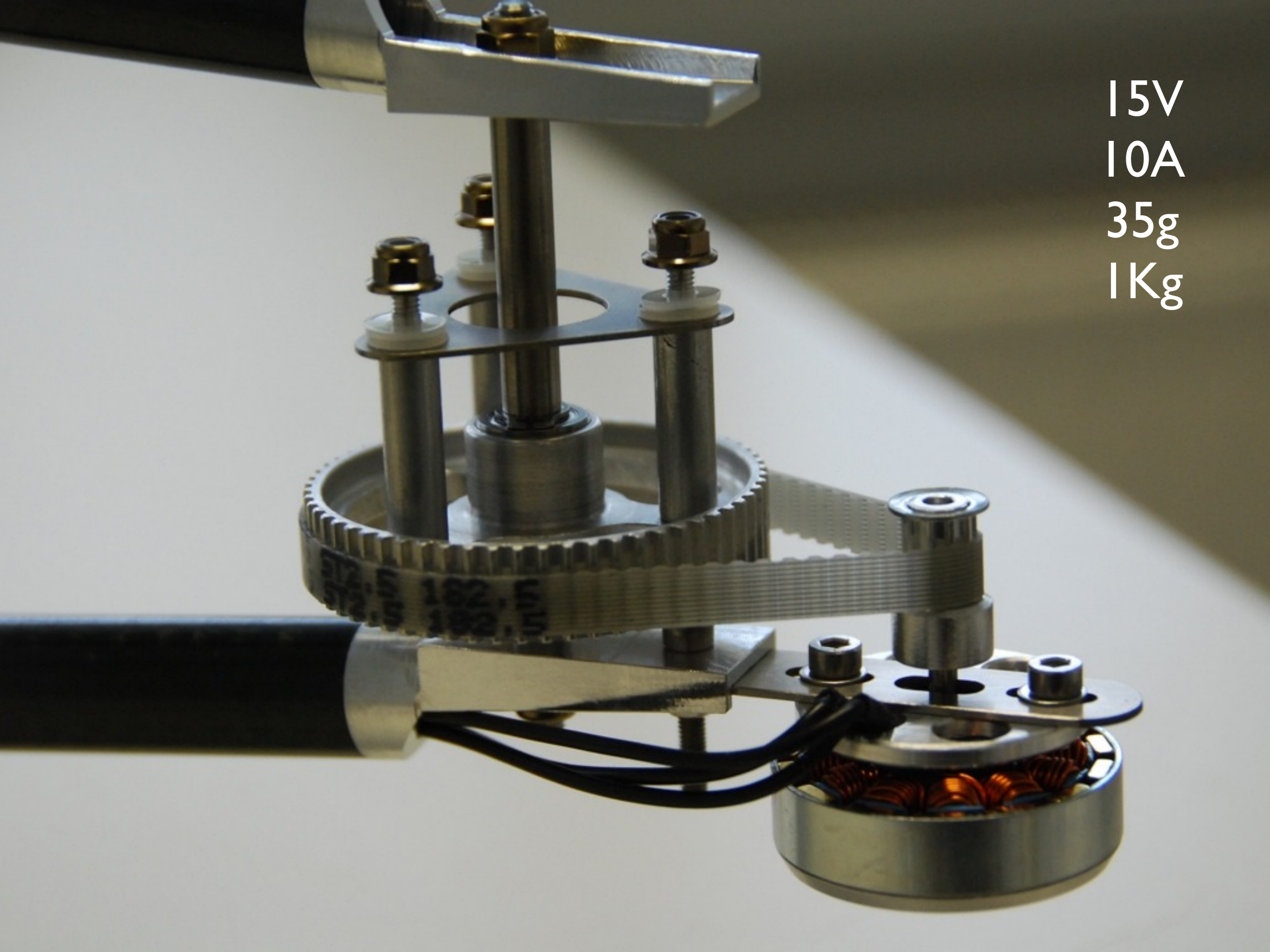
Minimal # of Different Parts



Minimal # of Different Parts

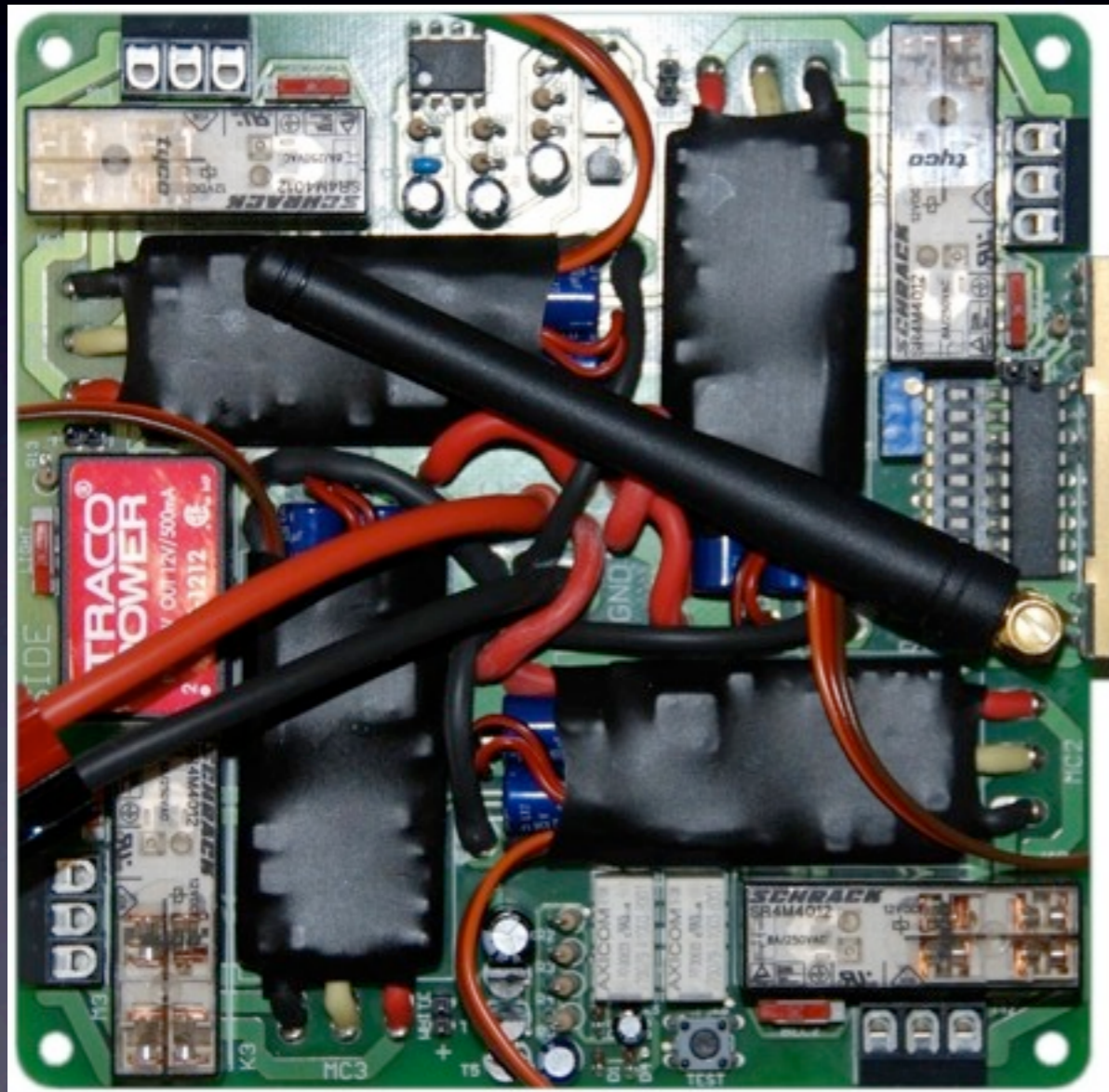


15V
10A
35g
1Kg

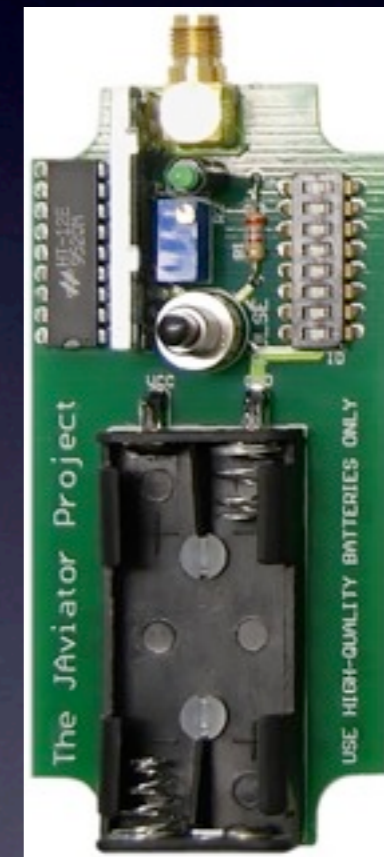




Custom Electronics

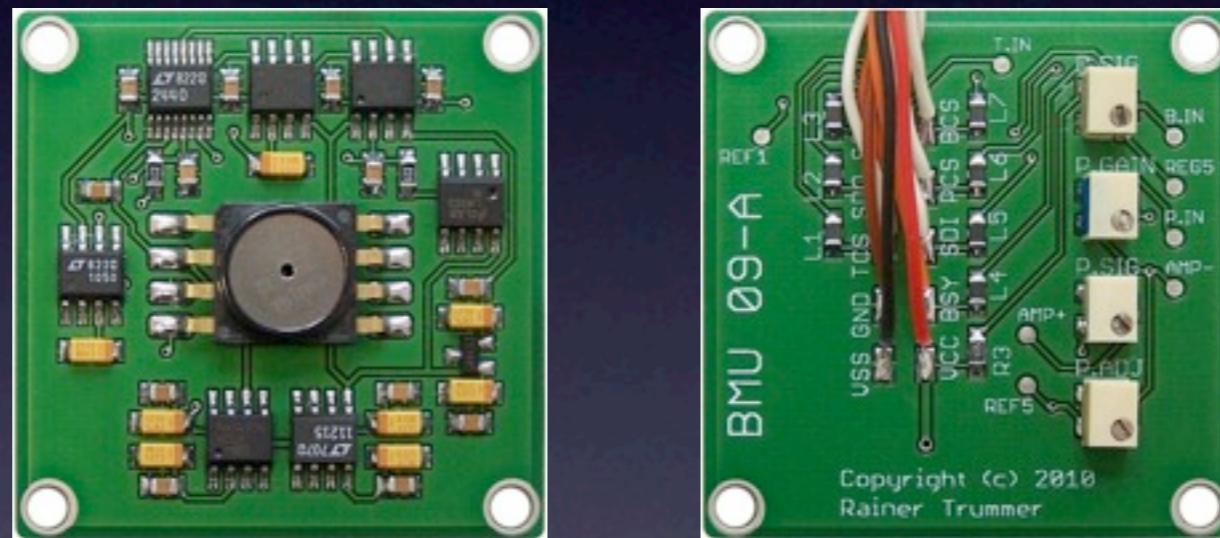


Power



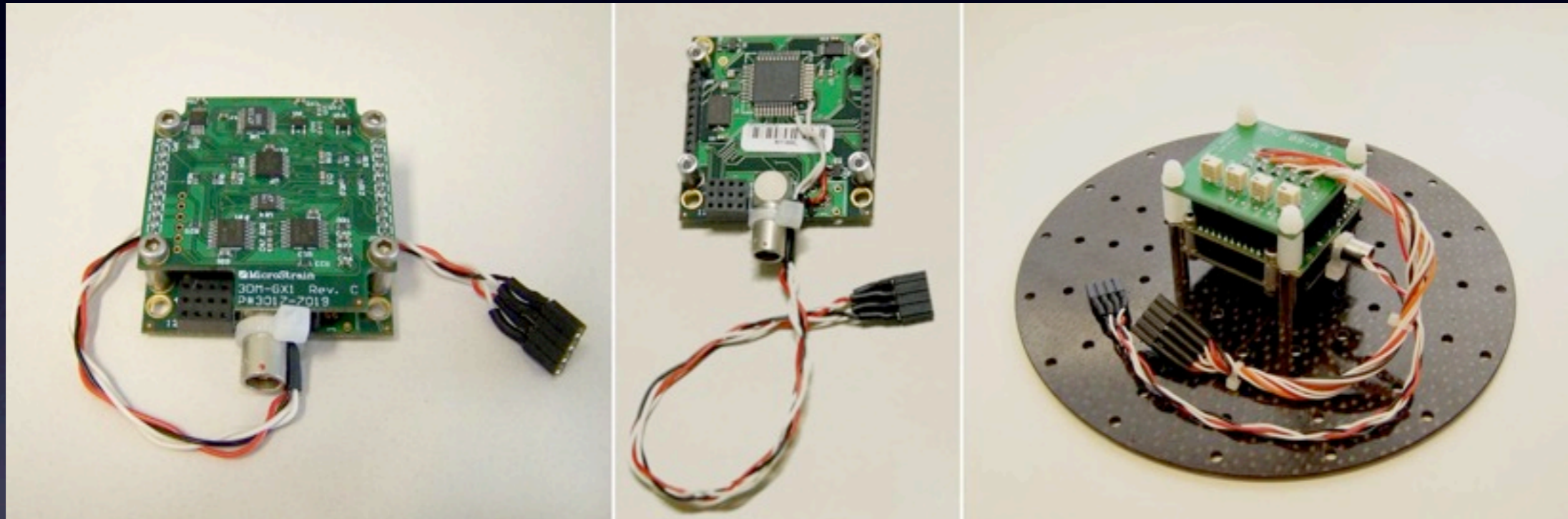
Remote

Custom Electronics



Barometer

Off-the-Shelf Stuff



Gyro

Off-the-Shelf Stuff



Ultrasonic

Off-the-Shelf Stuff

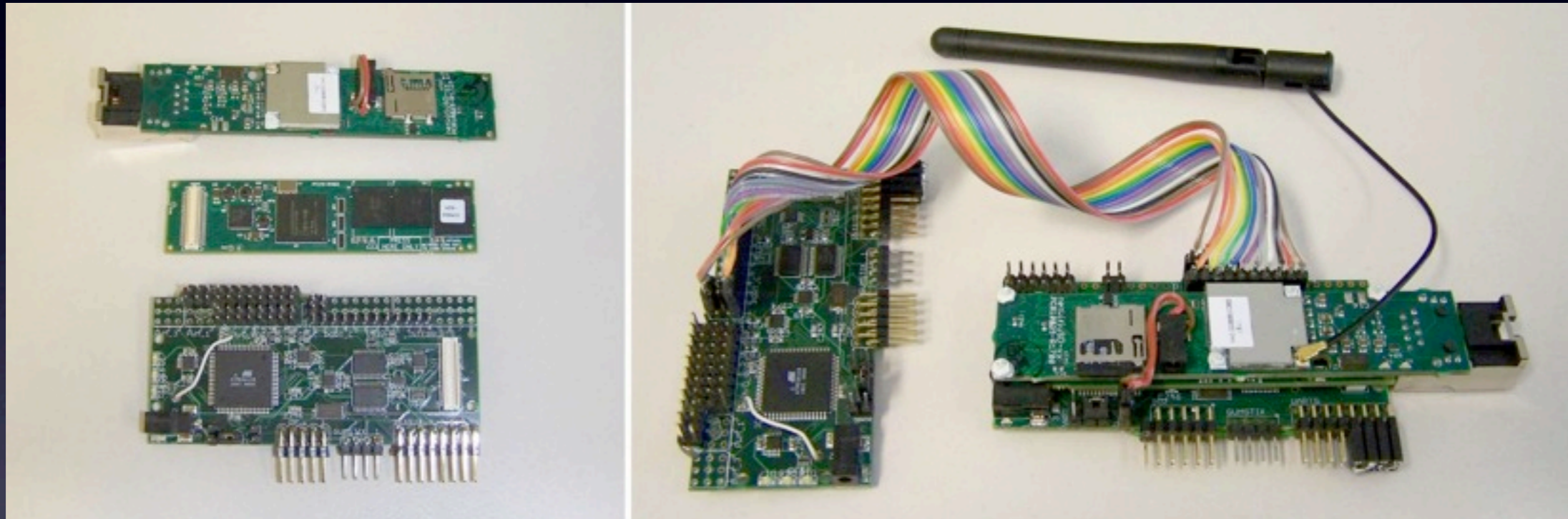


UWB RFID

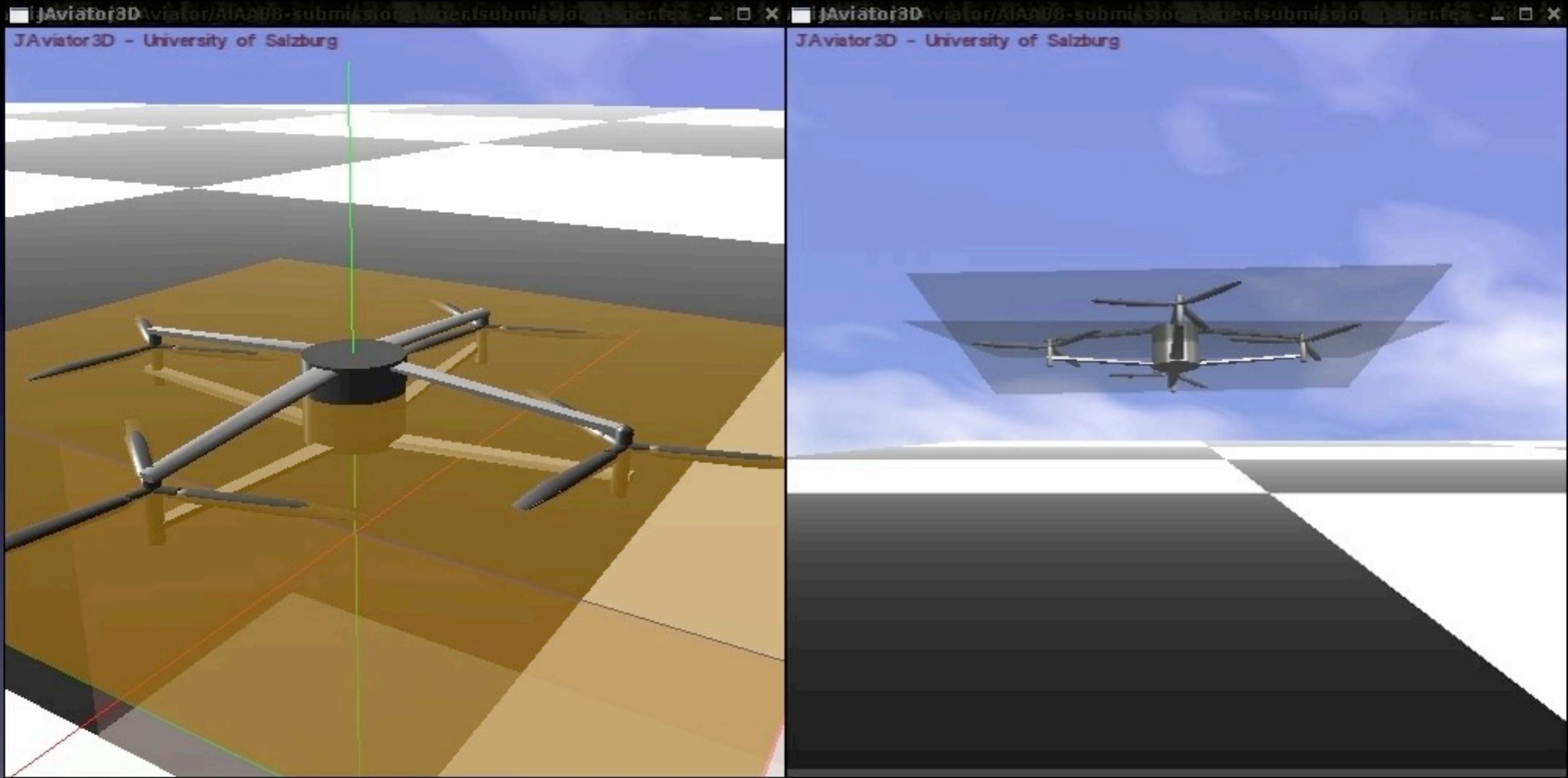


Laser

Off-the-Shelf Stuff



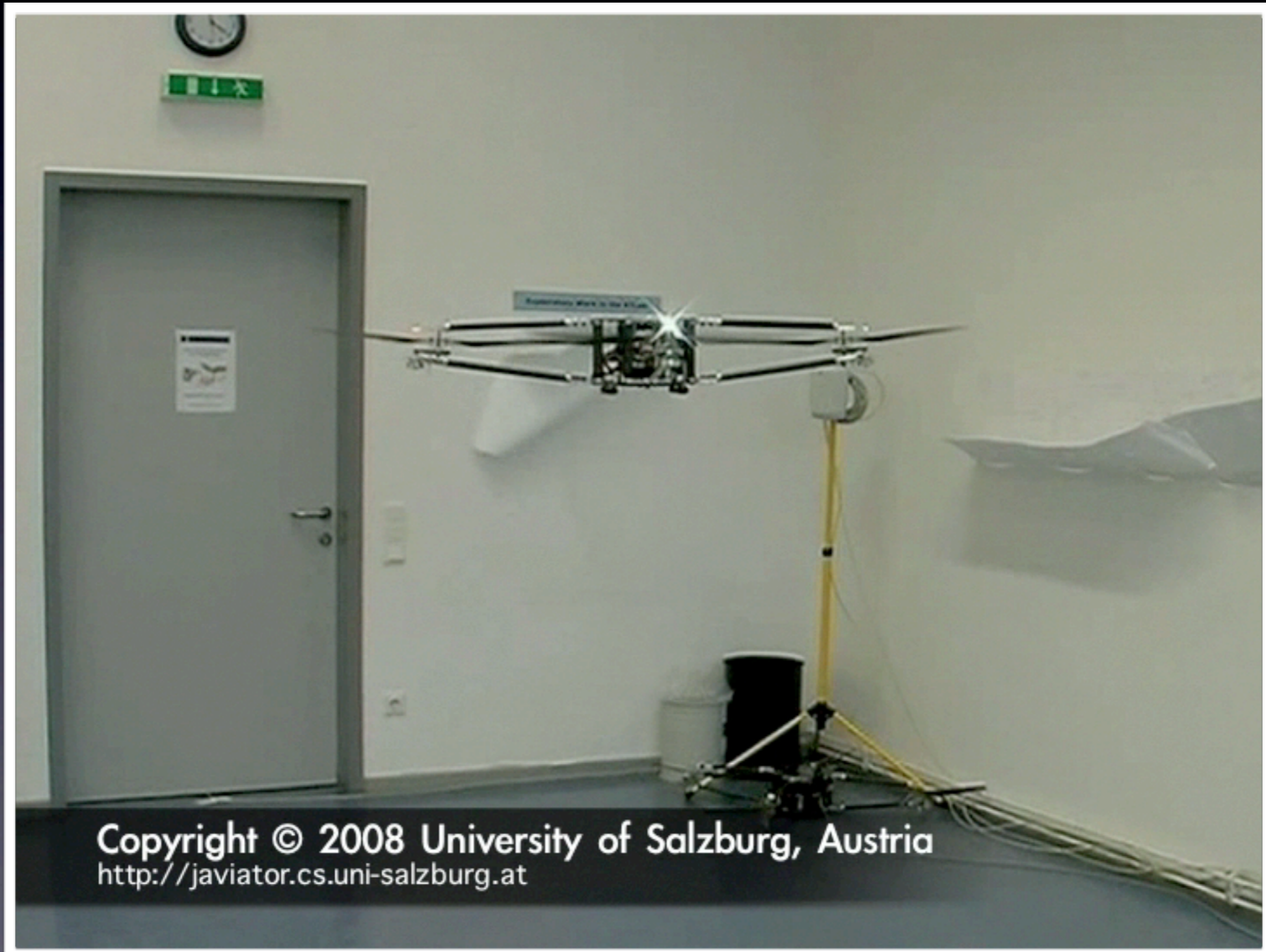
Gumstix





Indoor Flight STARMAC Controller

Indoor Flight STARMAC Controller



Copyright © 2008 University of Salzburg, Austria
<http://javiator.cs.uni-salzburg.at>

Outdoor Flight Salzburg Controller

Outdoor Flight Salzburg Controller



Copyright © 2008 University of Salzburg, Austria
<http://javiator.cs.uni-salzburg.at>

More Recent: Yawing

More Recent: Yawing



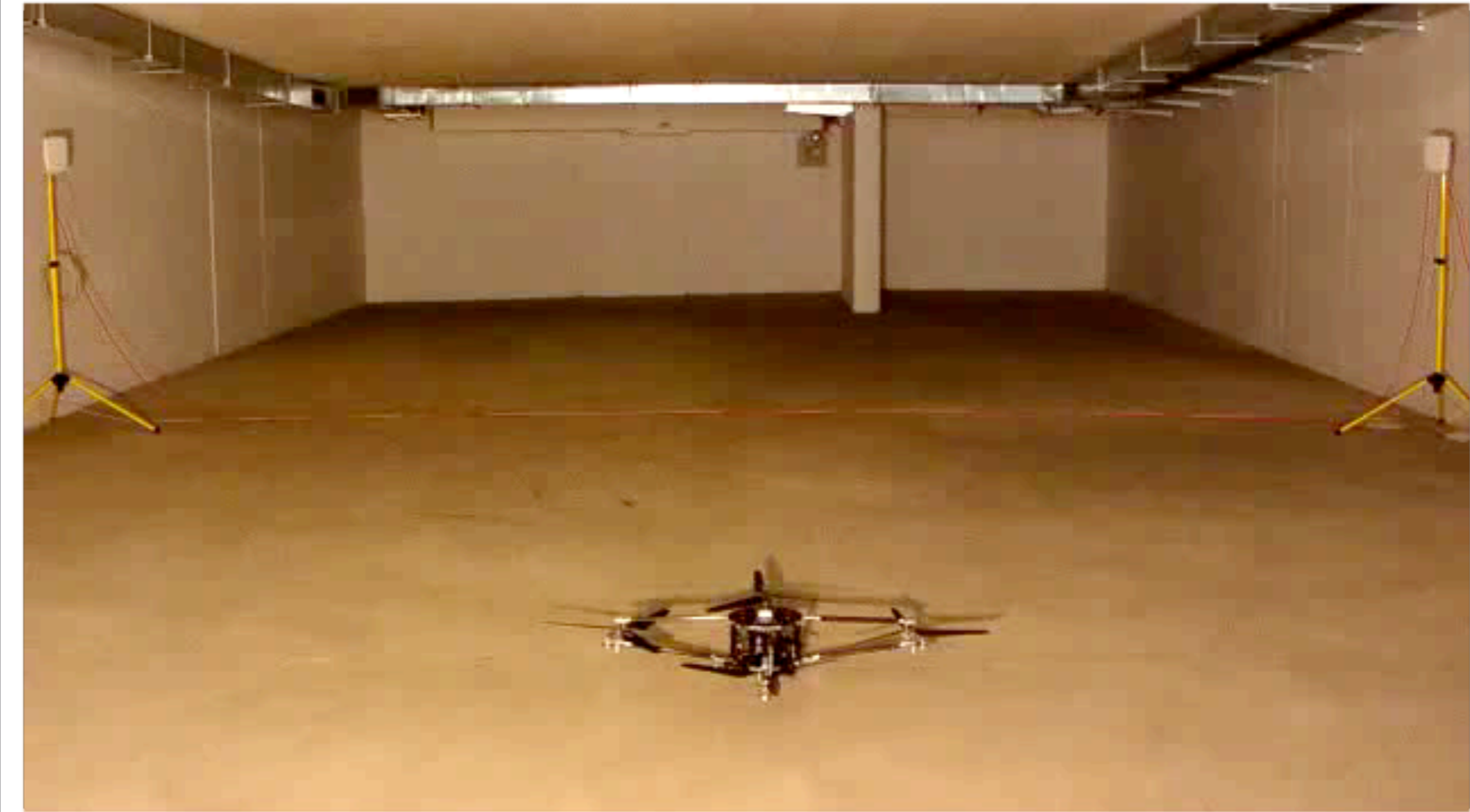
Oops

Oops



Autonomous

Autonomous



A Cyber-Physical Server



- IP address
- location

A Cyber-Physical Server



- IP address
- location
- capabilities

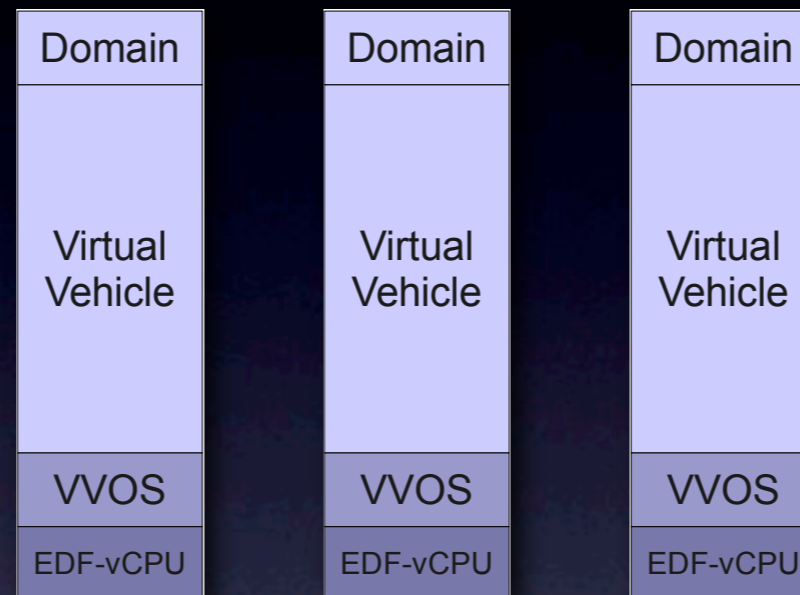
A Cyber-Physical Server



- IP address
- location
- capabilities
- motion

A Cyber-Physical Server

- IP address
- location
- capabilities
- motion



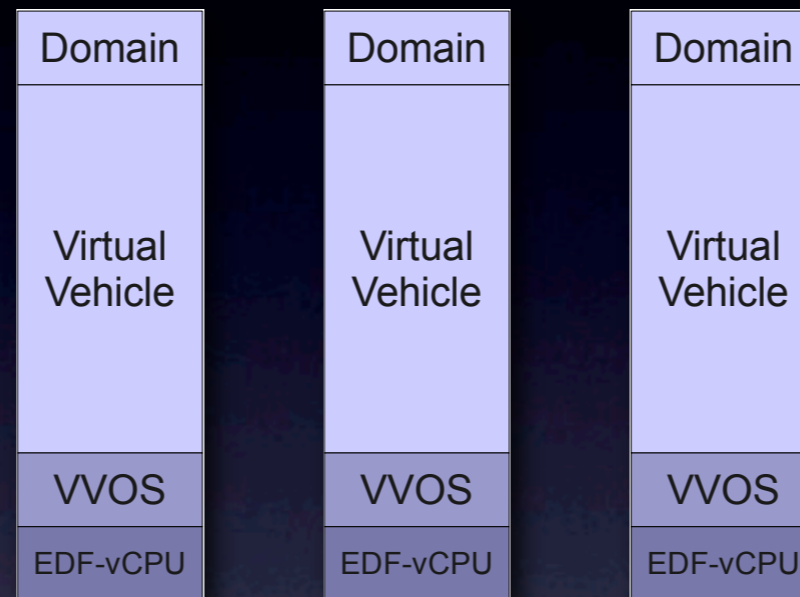
- IP address
- location
- capabilities
- motion



- IP address
- location
- capabilities
- motion

A Cyber-Physical Server

- IP address
- location
- capabilities
- motion



- IP address
- location
- capabilities
- motion

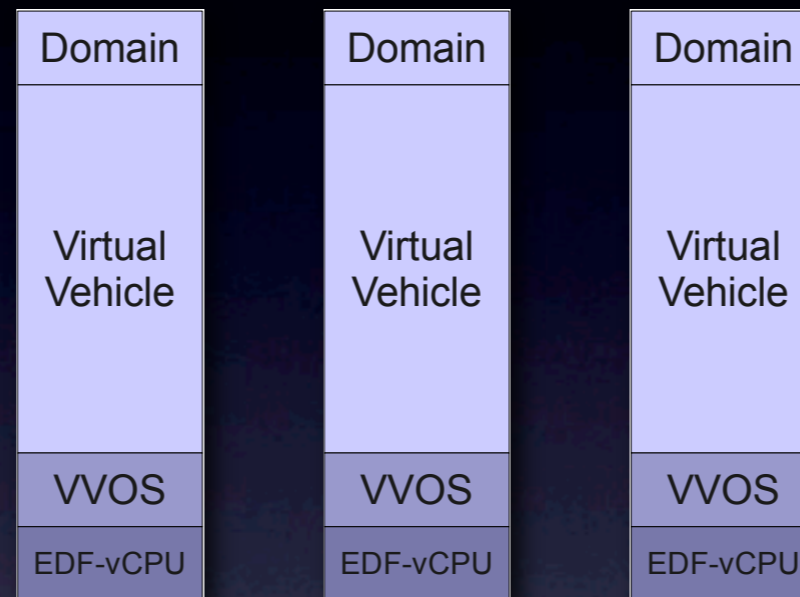


restricted

- IP address
- location
- capabilities
- motion

A Cyber-Physical Server

- IP address
- location
- capabilities
- motion



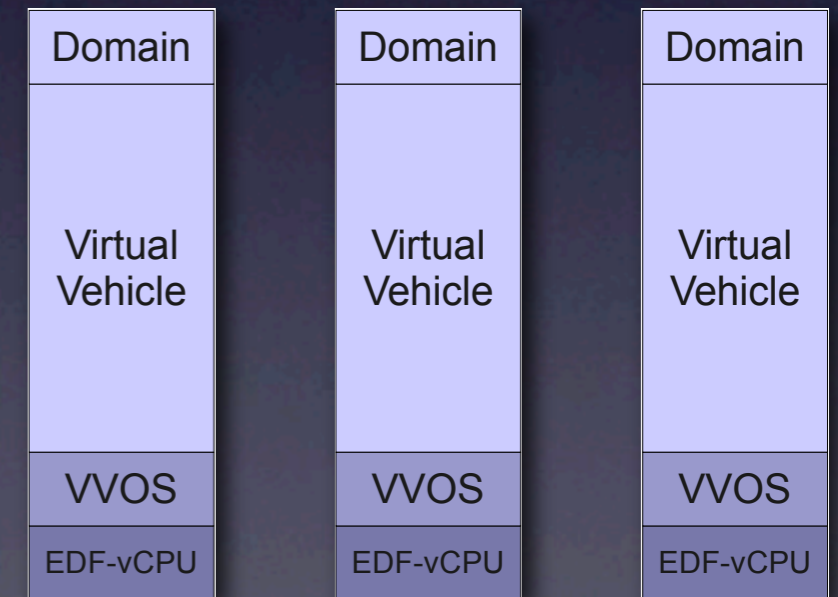
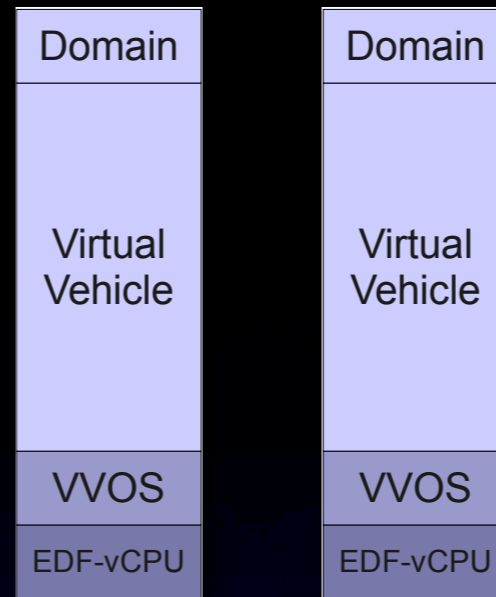
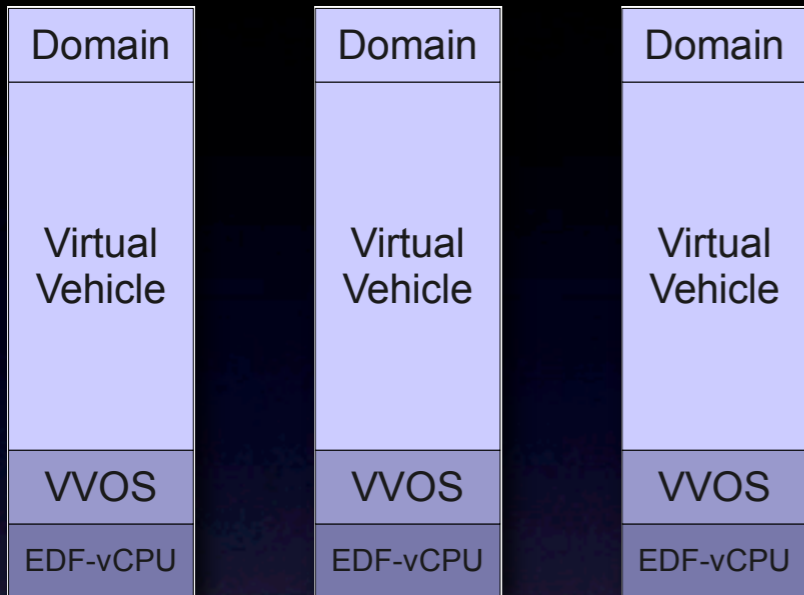
- IP address
- location
- capabilities
- motion



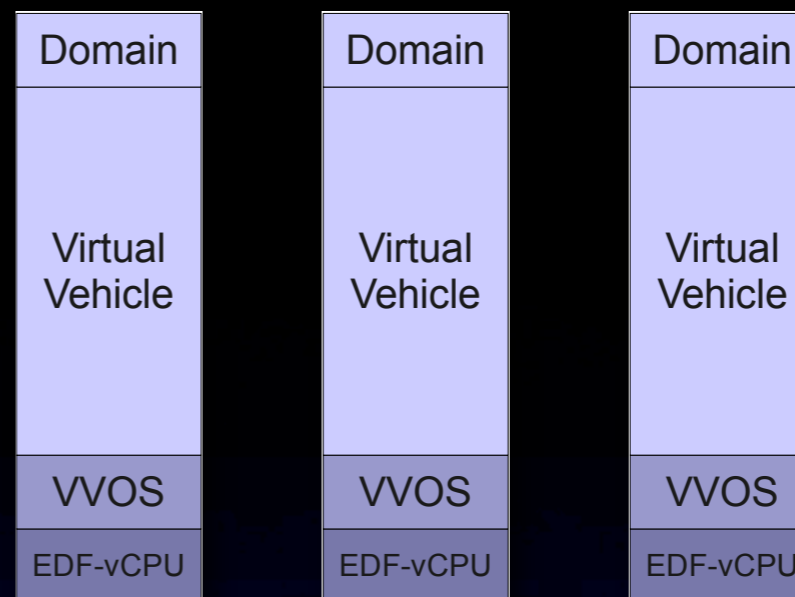
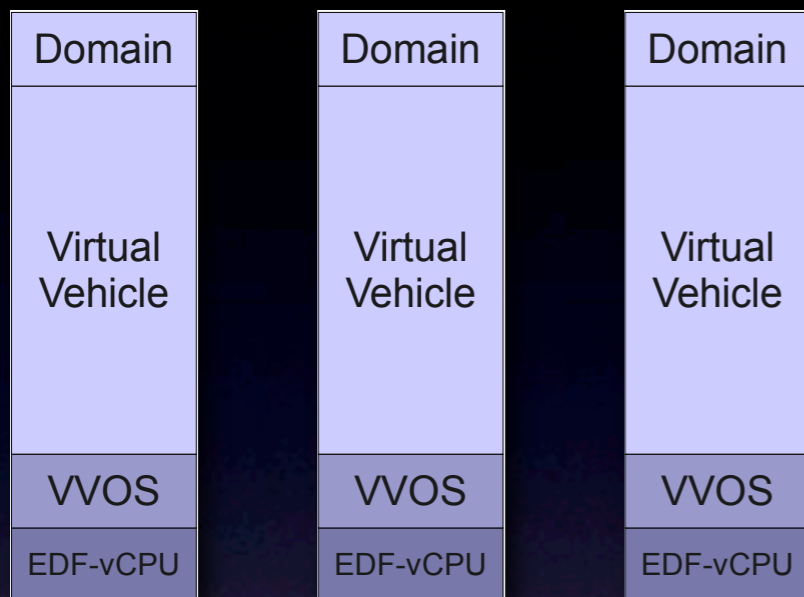
restricted

idealized

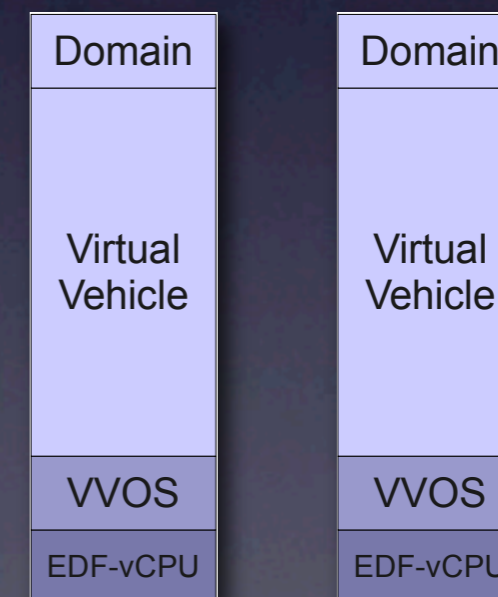
- IP address
- location
- capabilities
- motion



A Cyber-Physical Cloud [HotCloud 2010]



migration
=
flying



A Cyber-Physical Cloud [HotCloud 2010]

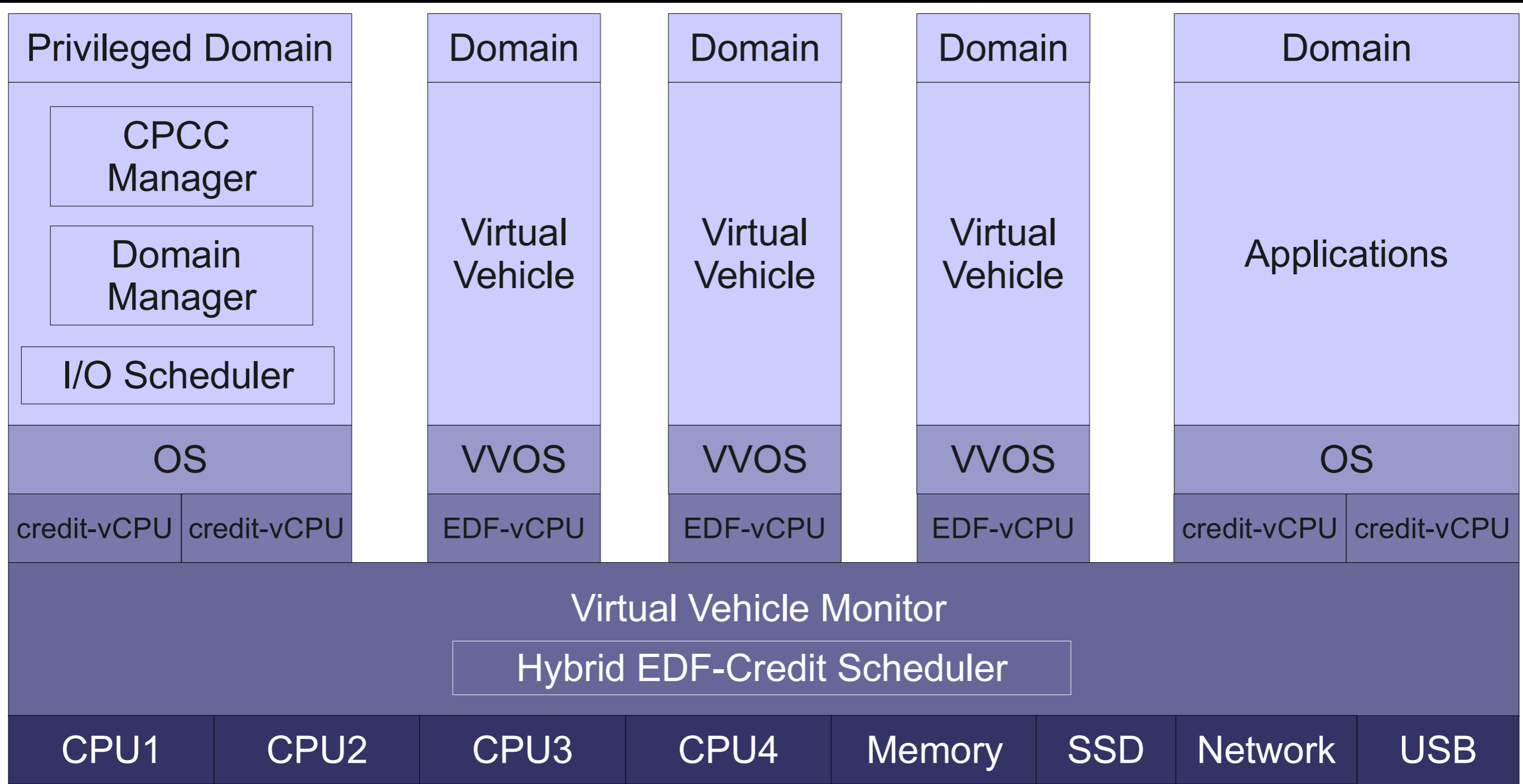
Goals

- **Multi-provider** (10s):
 - heterogeneous operations
- **Multi-vehicle** (100s):
 - heterogeneous systems
- **Multi-task** (1000s):
 - heterogeneous missions

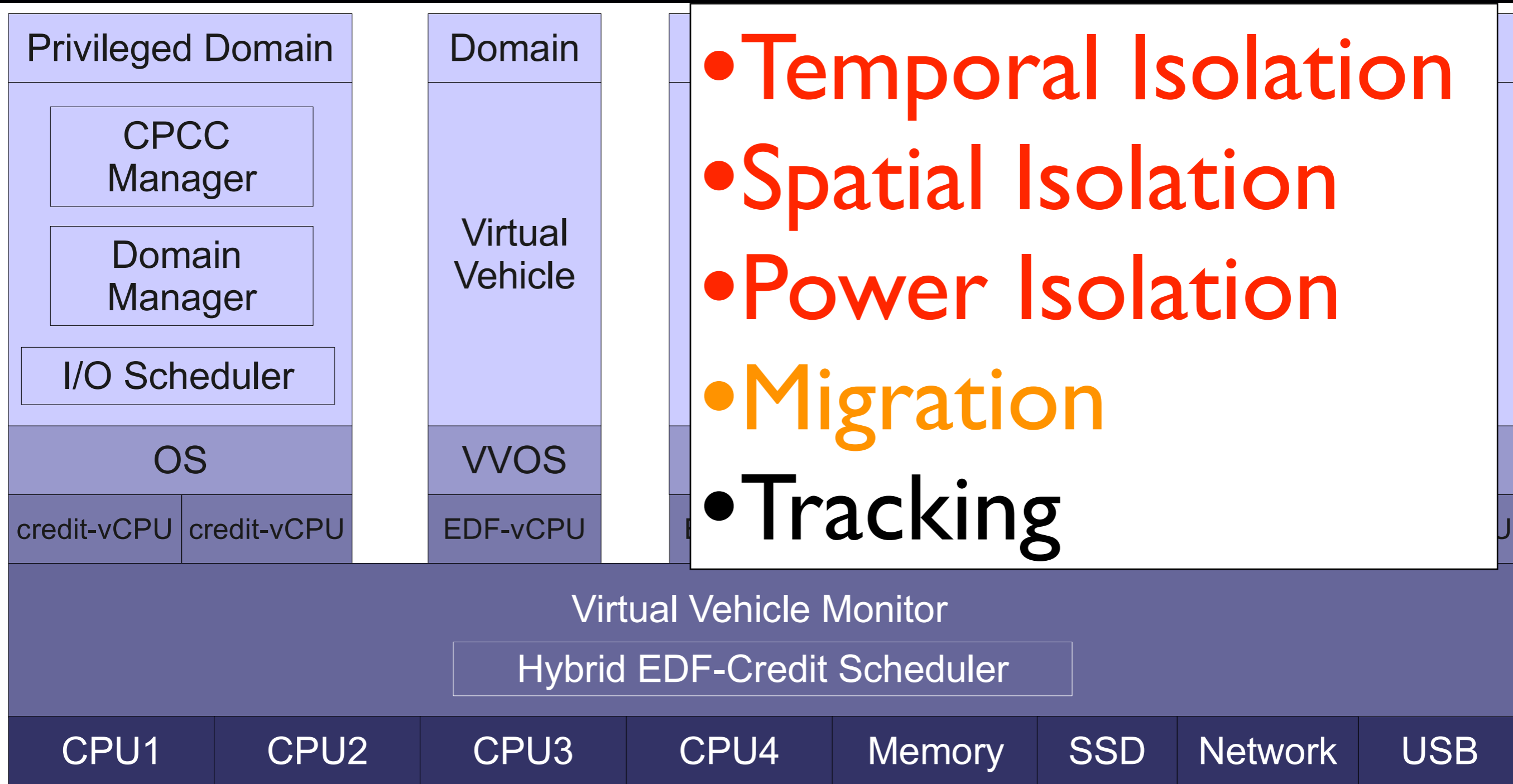
High-Level Challenges

- Virtualization **Infrastructure**
 - ▶ Salzburg
- Collaborative **Control**
 - ▶ Berkeley
- Programming **Language**
 - ▶ Berkeley, Salzburg

Virtualization Infrastructure



Virtualization Infrastructure



There is a
fundamental trade-off
between
quality and cost
of
time, space, power
isolation

Time

- **quality**: response time jitter
- **cost**: scheduling overhead

Time

- **quality**: response time jitter
- **cost**: scheduling overhead

Space

- **quality**: fragmentation jitter
- **cost**: management overhead

Time

- **quality**: response time jitter
- **cost**: scheduling overhead

Space

- **quality**: fragmentation jitter
- **cost**: management overhead

Power

- **quality**: power consumption jitter
- **cost**: total power consumption

Time

Variable-Bandwidth Servers (VBS)
[SIES09]

VBS Overhead Accounting
[RTAS10]

Space

Compact-fit
[USENIX ATC08]

Short-term Memory
[Submitted 11]

Power

Power-aware VBS
[EMSOFT10]

The Power of Isolation
[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]

VBS Overhead Accounting
[RTAS10]

Compact-fit

[USENIX ATC08]

Space

Short-term Memory

[Submitted 11]

Power-aware VBS

[EMSOFT10]

Power

The Power of Isolation

[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]
- account **overhead** in utilization and/or response times [RTAS10]

Space

Compact-fit
[USENIX ATC08]

Short-term Memory
[Submitted 11]

Power

Power-aware VBS
[EMSOFT10]

The Power of Isolation
[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]
- account **overhead** in utilization and/or response times [RTAS10]
- **time-** and **space-predictable** malloc and free [USENIX ATC08]

Space

Short-term Memory

[Submitted 11]

Power-aware VBS

[EMSOFT10]

Power

The Power of Isolation

[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]
- account **overhead** in utilization and/or response times [RTAS10]
- **time-** and **space-predictable** malloc and free [USENIX ATC08]
- **refresh** needed rather than deallocate not-needed objects [Submitted 11]

Space

Power

Power-aware VBS
[EMSOFT10]

The Power of Isolation
[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]
- account **overhead** in utilization and/or response times [RTAS10]

Space

- **time-** and **space-predictable** malloc and free [USENIX ATC08]
- **refresh** needed rather than deallocate not-needed objects [Submitted 11]

Power

- save power **while** maintaining response times [EMSOFT10]

The Power of Isolation
[Submitted 11]

Time

- **per-process** lower/upper bounds on response times [SIES09]
- account **overhead** in utilization and/or response times [RTAS10]

Space

- **time-** and **space-predictable** malloc and free [USENIX ATC08]
- **refresh** needed rather than deallocate not-needed objects [Submitted 11]

Power

- save power **while** maintaining response times [EMSOFT10]
- per-process lower/upper bounds on **power consumption** [Submitted 11]

The trade-off
between
isolation **quality** and **cost**
extends to
scalability
on multicore systems

Multicore

- many concurrent data structures **do not scale**,
e.g. a concurrent FIFO queue for scheduling

Multicore

- many concurrent data structures **do not scale**, e.g. a concurrent FIFO queue for scheduling
- **linearizability** makes a concurrent FIFO queue return the “oldest” element but **limits** scalability

Multicore

- many concurrent data structures **do not scale**, e.g. a concurrent FIFO queue for scheduling
- **linearizability** makes a concurrent FIFO queue return the “oldest” element but **limits** scalability
- **k-linearizability** allows it to **scale** but also return up to the “**k**-oldest” element

Multicore

- many concurrent data structures **do not scale**, e.g. a concurrent FIFO queue for scheduling
- **linearizability** makes a concurrent FIFO queue return the “oldest” element but **limits** scalability
- **k-linearizability** allows it to **scale** but also return up to the “**k**-oldest” element
- trades off **semantics** and **scalability**

Multicore

- many concurrent data structures **do not scale**, e.g. a concurrent FIFO queue for scheduling
- **linearizability** makes a concurrent FIFO queue return the “oldest” element but **limits** scalability
- **k-linearizability** allows it to **scale** but also return up to the “**k**-oldest” element
- trades off **semantics** and **scalability**
- can be done **probabilistically**

Multicore

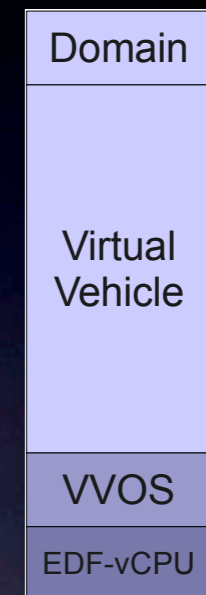
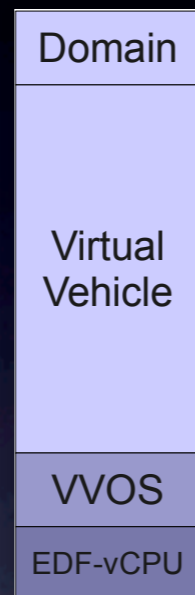
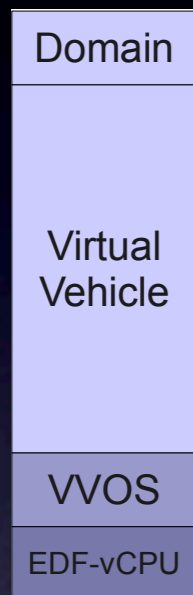
- many concurrent data structures **do not scale**, e.g. a concurrent FIFO queue for scheduling
- **linearizability** makes a concurrent FIFO queue return the “oldest” element but **limits** scalability
- **k-linearizability** allows it to **scale** but also return up to the “**k**-oldest” element
- trades off **semantics** and **scalability**
- can be done **probabilistically**

Non-linearizable Computing

Breaks the Scalability Barrier [Submitted I I]

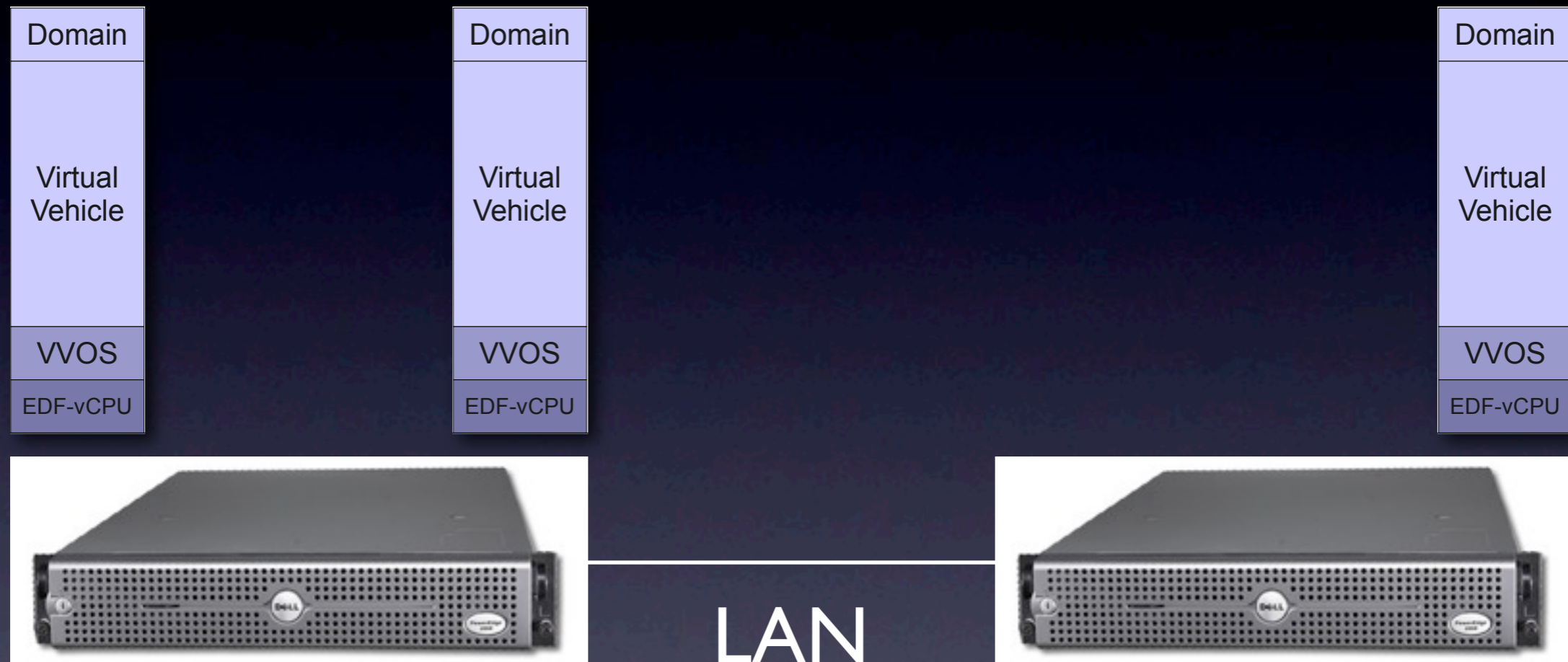
Virtual Vehicle Demo

by Florian Landolt and Andreas Rottmann



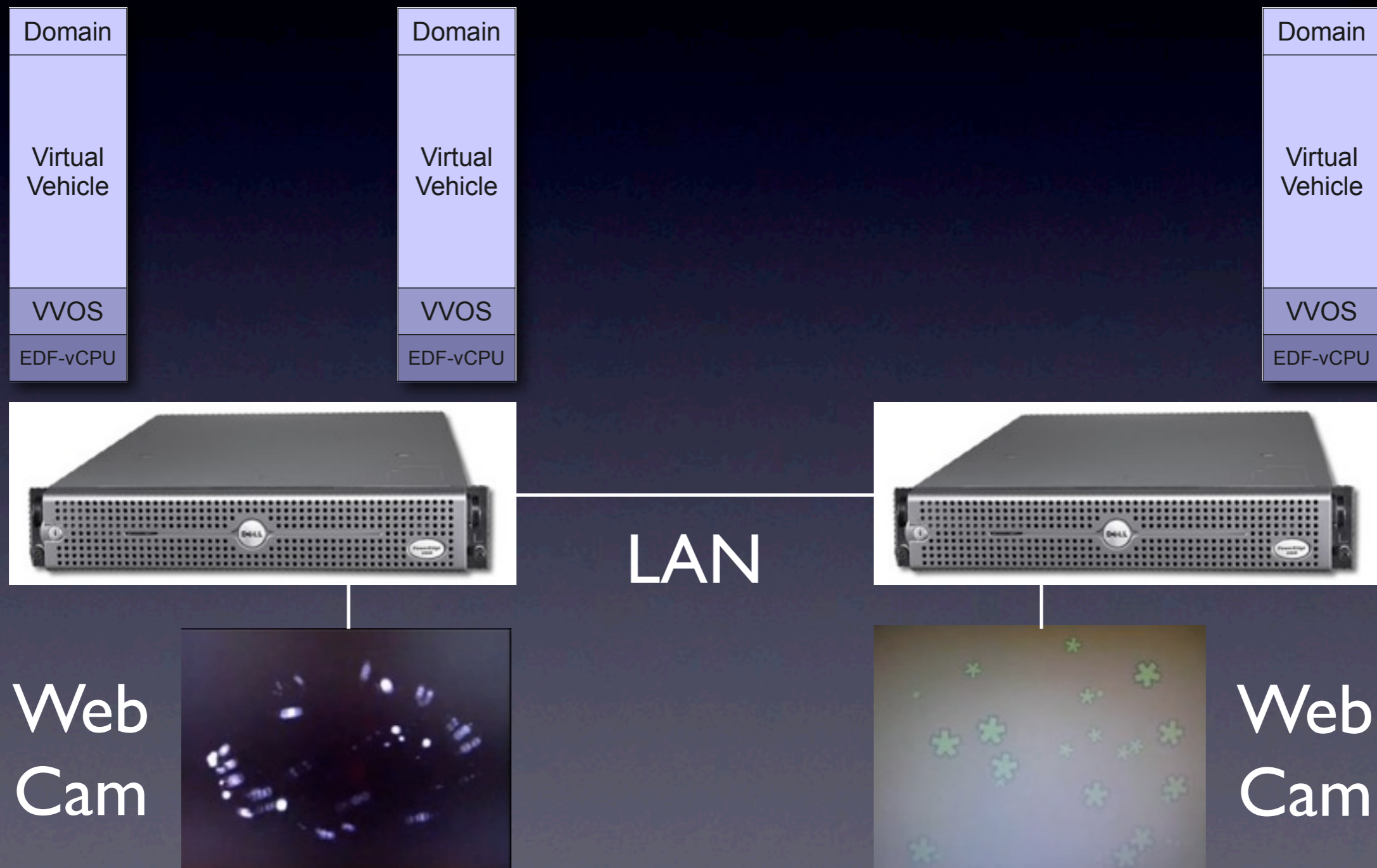
Virtual Vehicle Demo

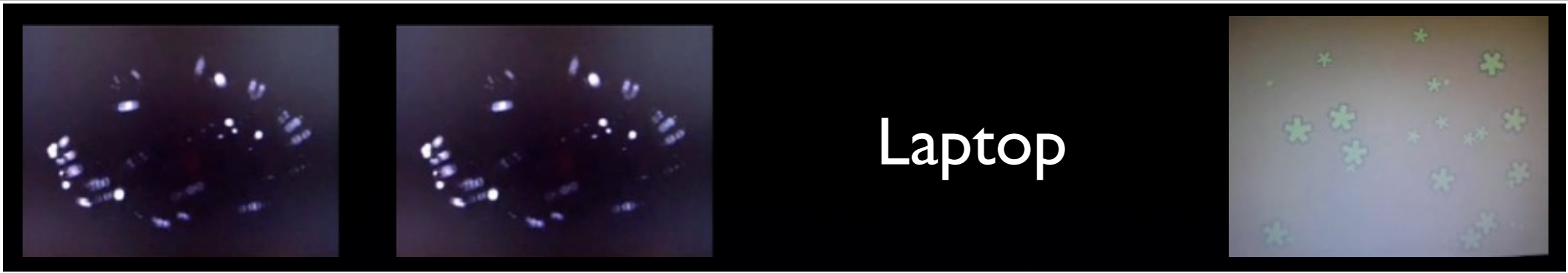
by Florian Landolt and Andreas Rottmann



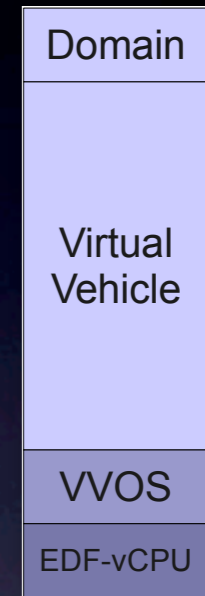
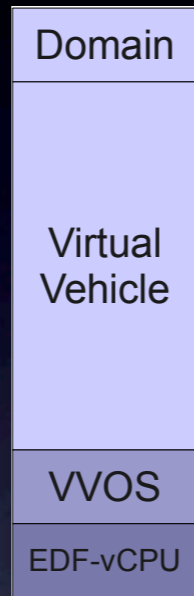
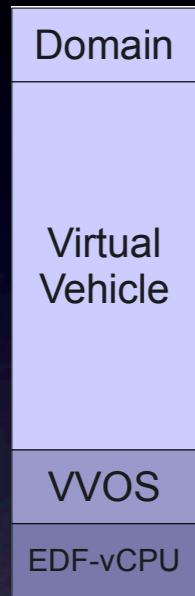
Virtual Vehicle Demo

by Florian Landolt and Andreas Rottmann





Laptop



LAN

Web
Cam

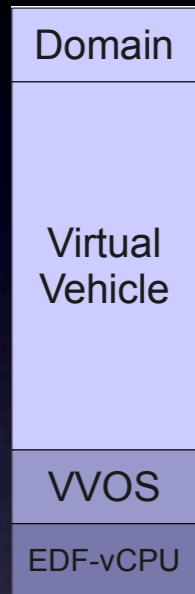


Web
Cam

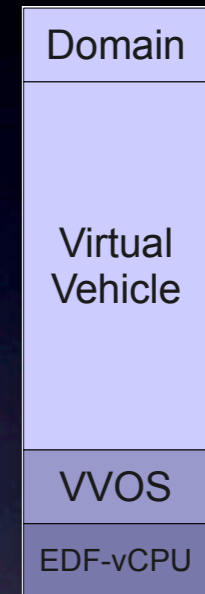
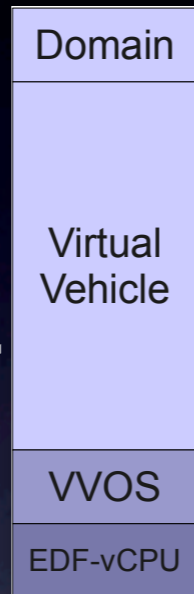




Laptop



Multicast



LAN

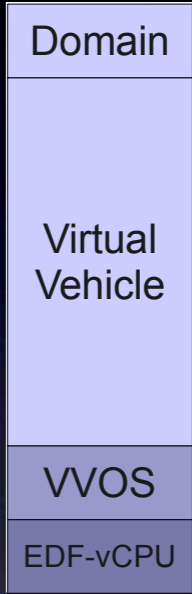
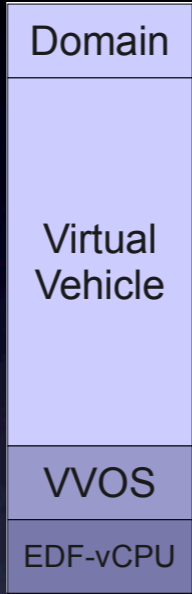
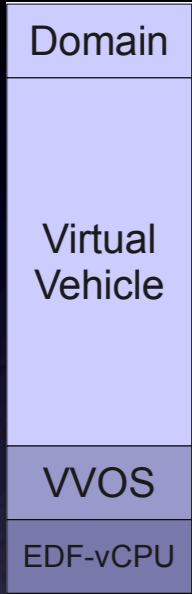
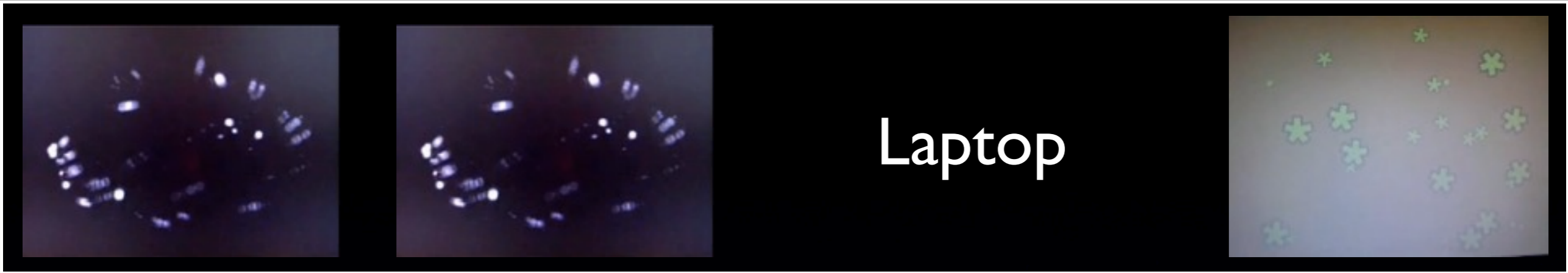


Web Cam



Web Cam





Migration



LAN

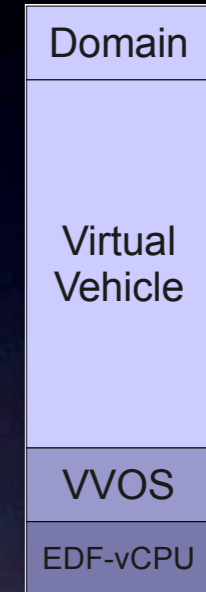
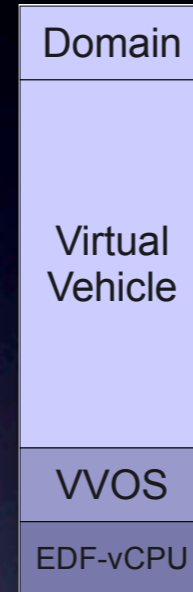
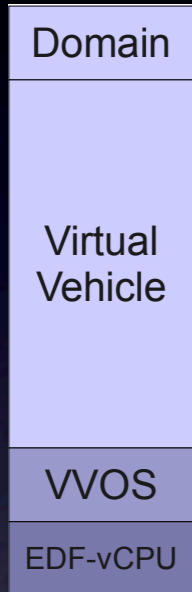
Web Cam



Web Cam



Laptop



LAN

Web
Cam

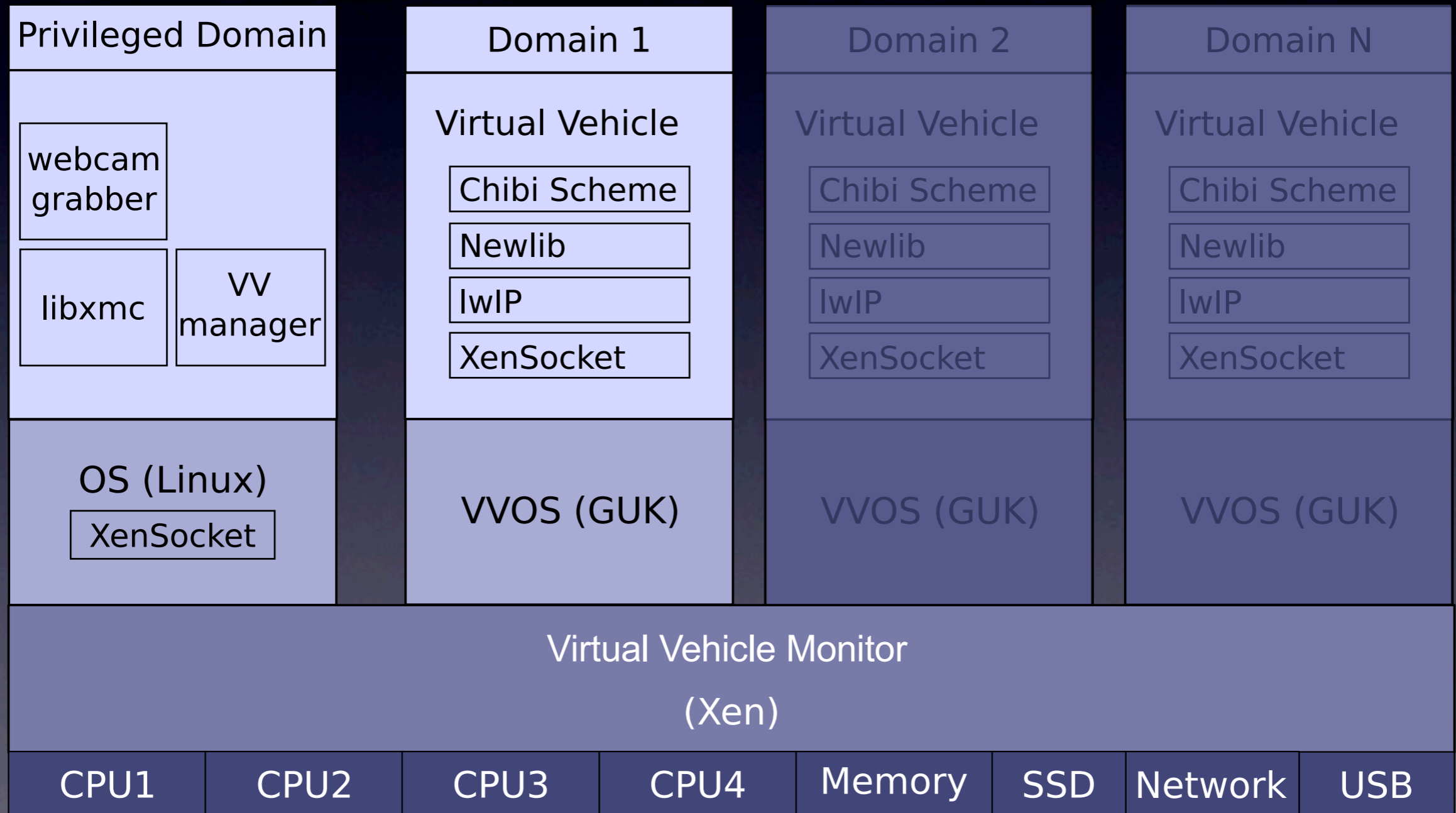


Web
Cam



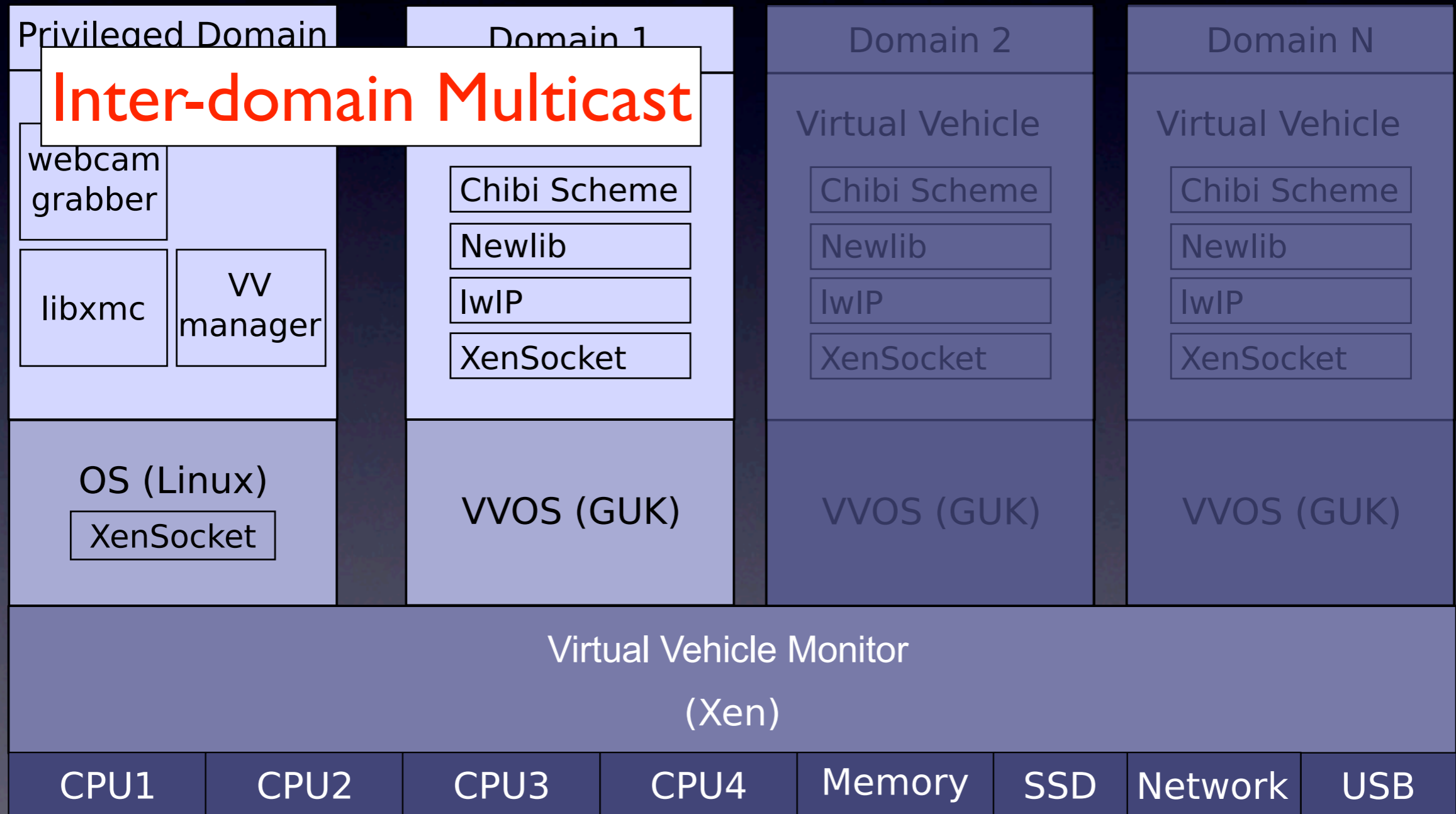
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



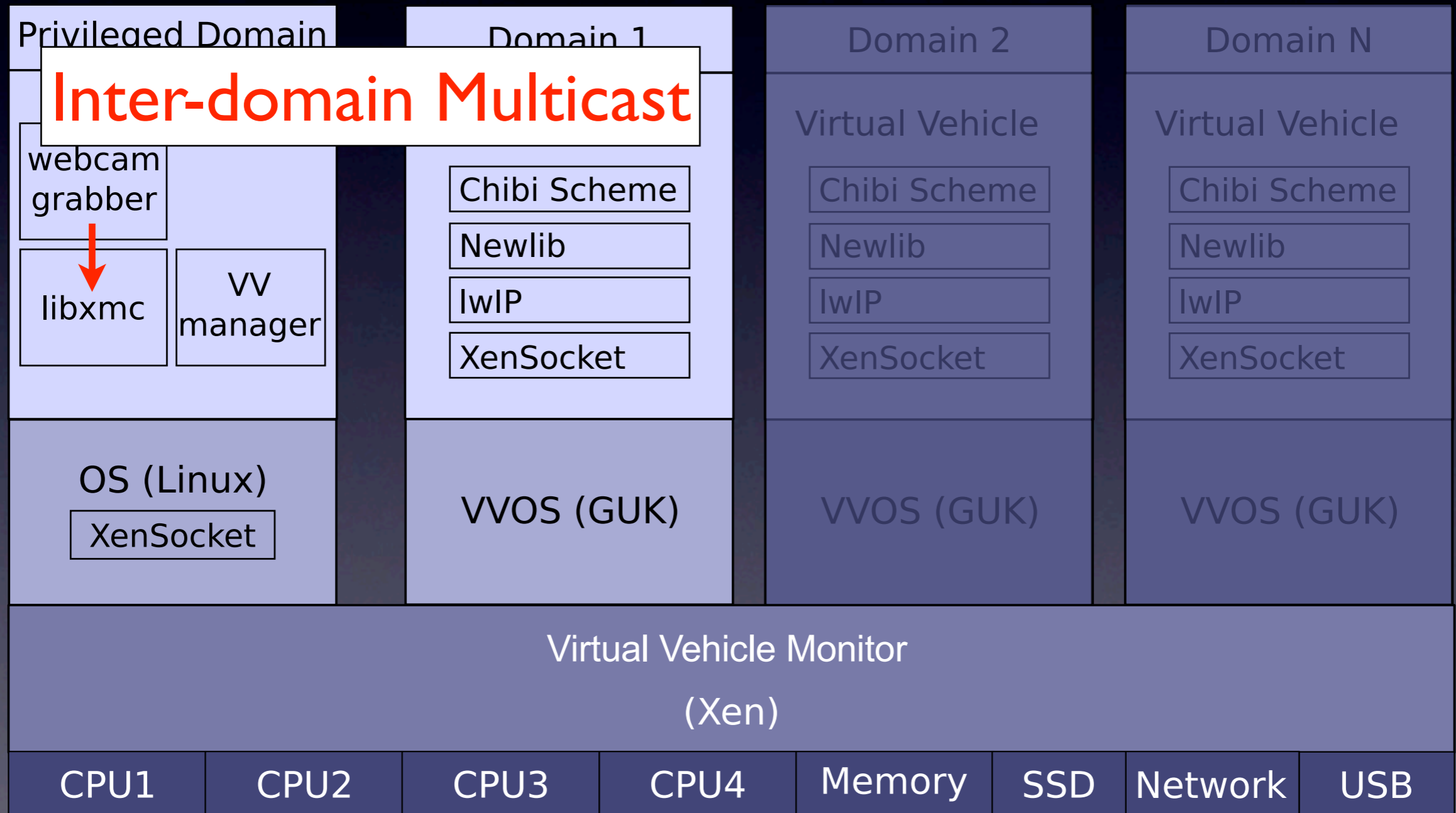
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



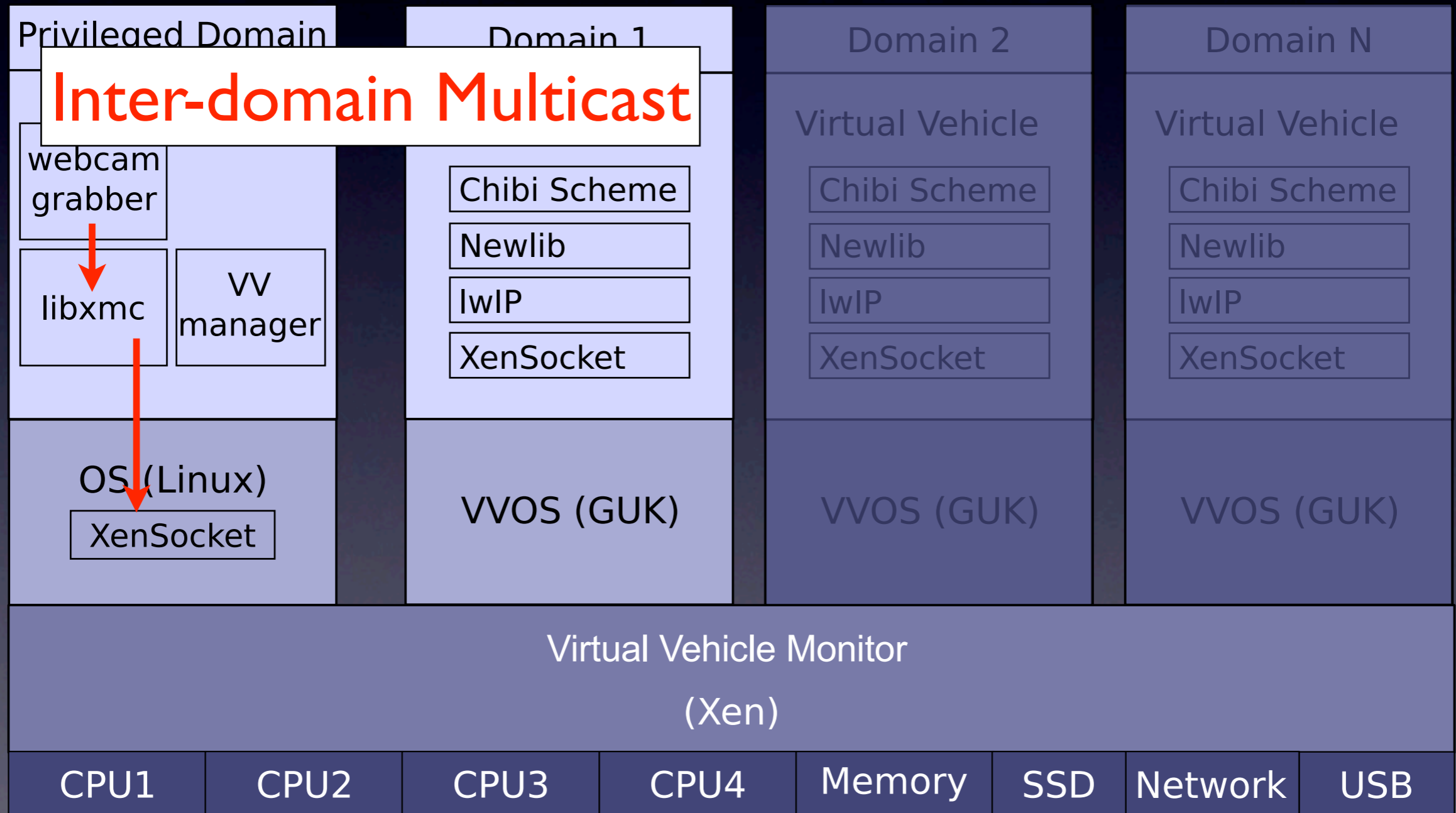
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



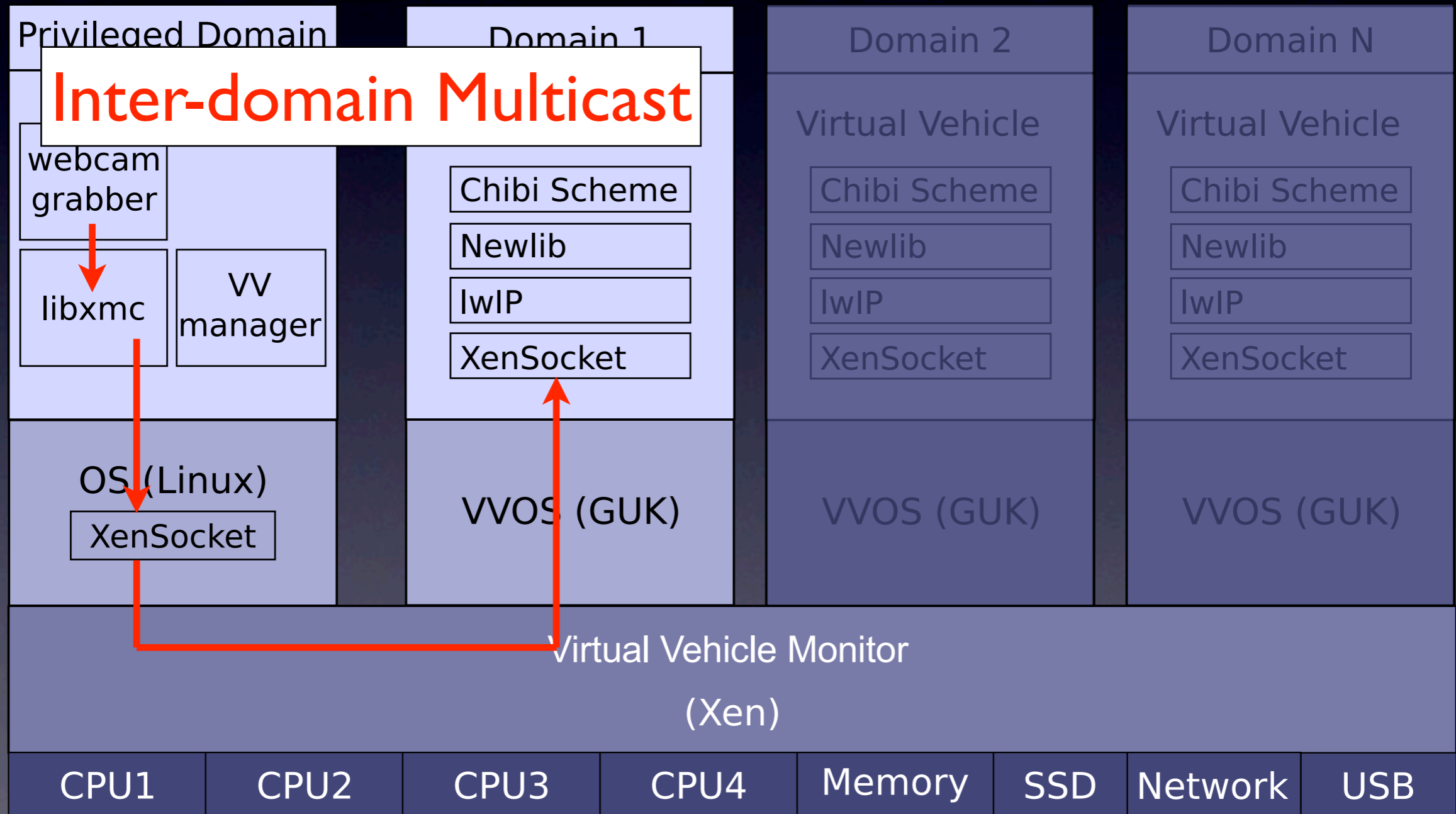
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



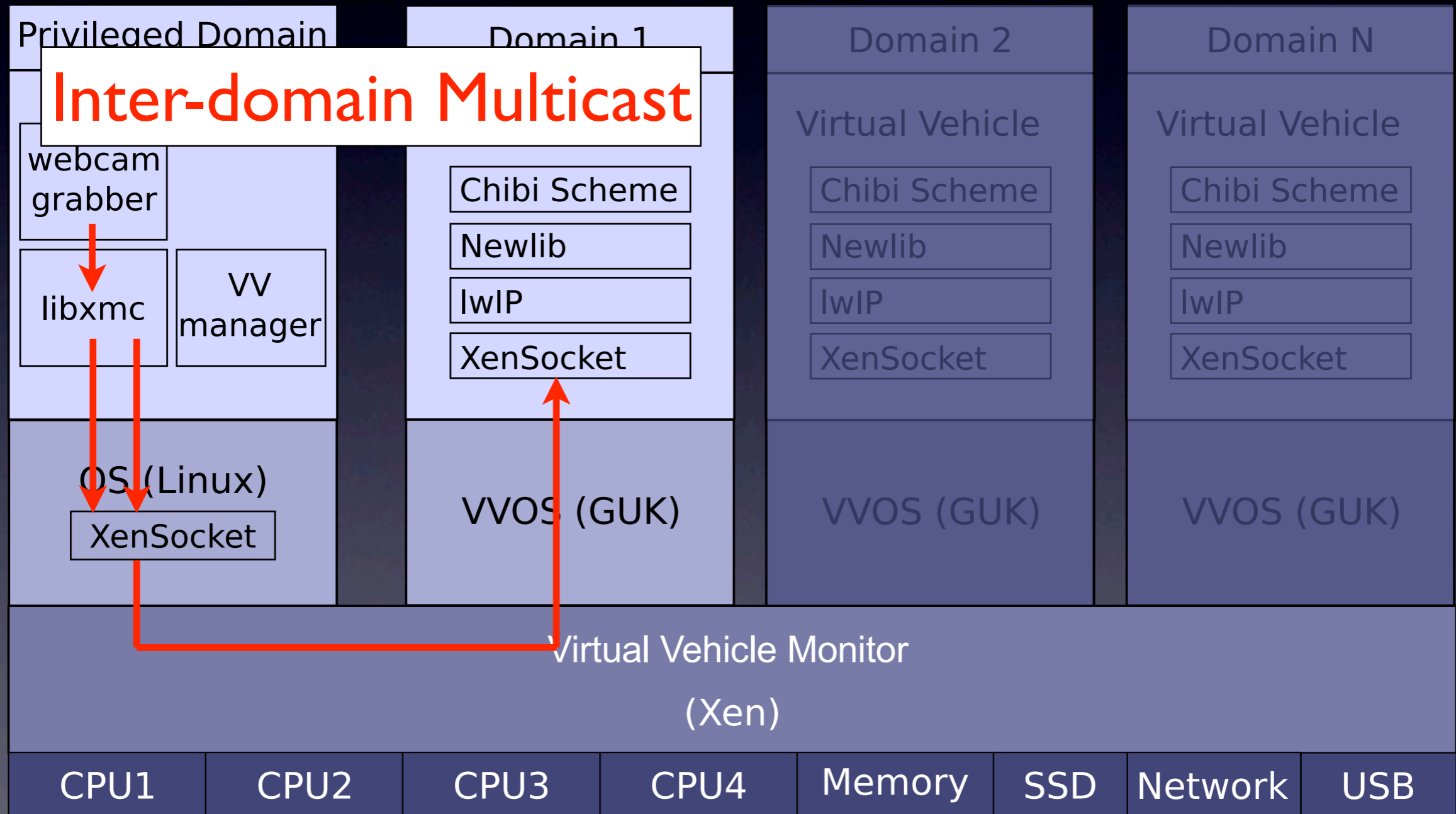
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



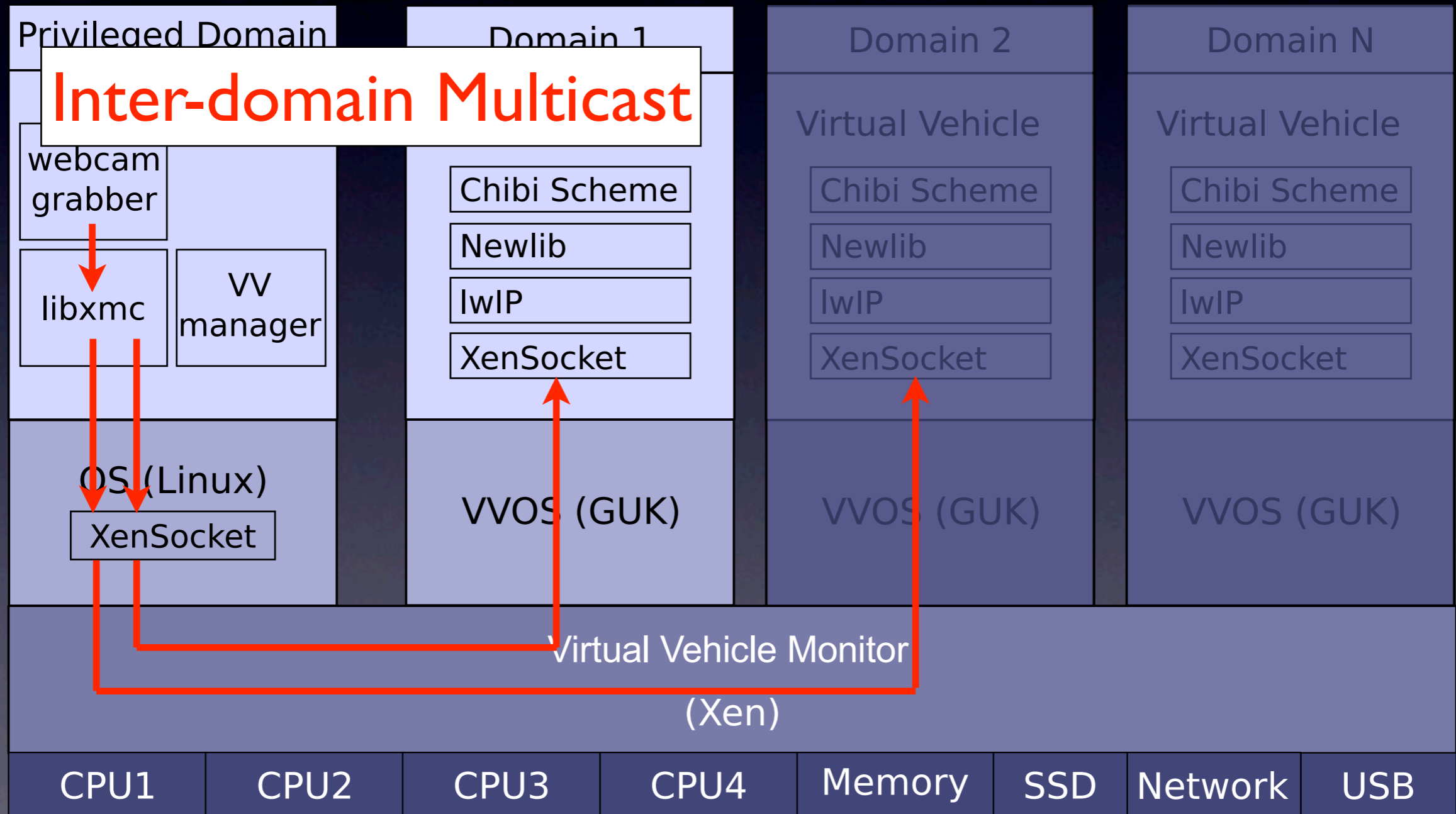
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



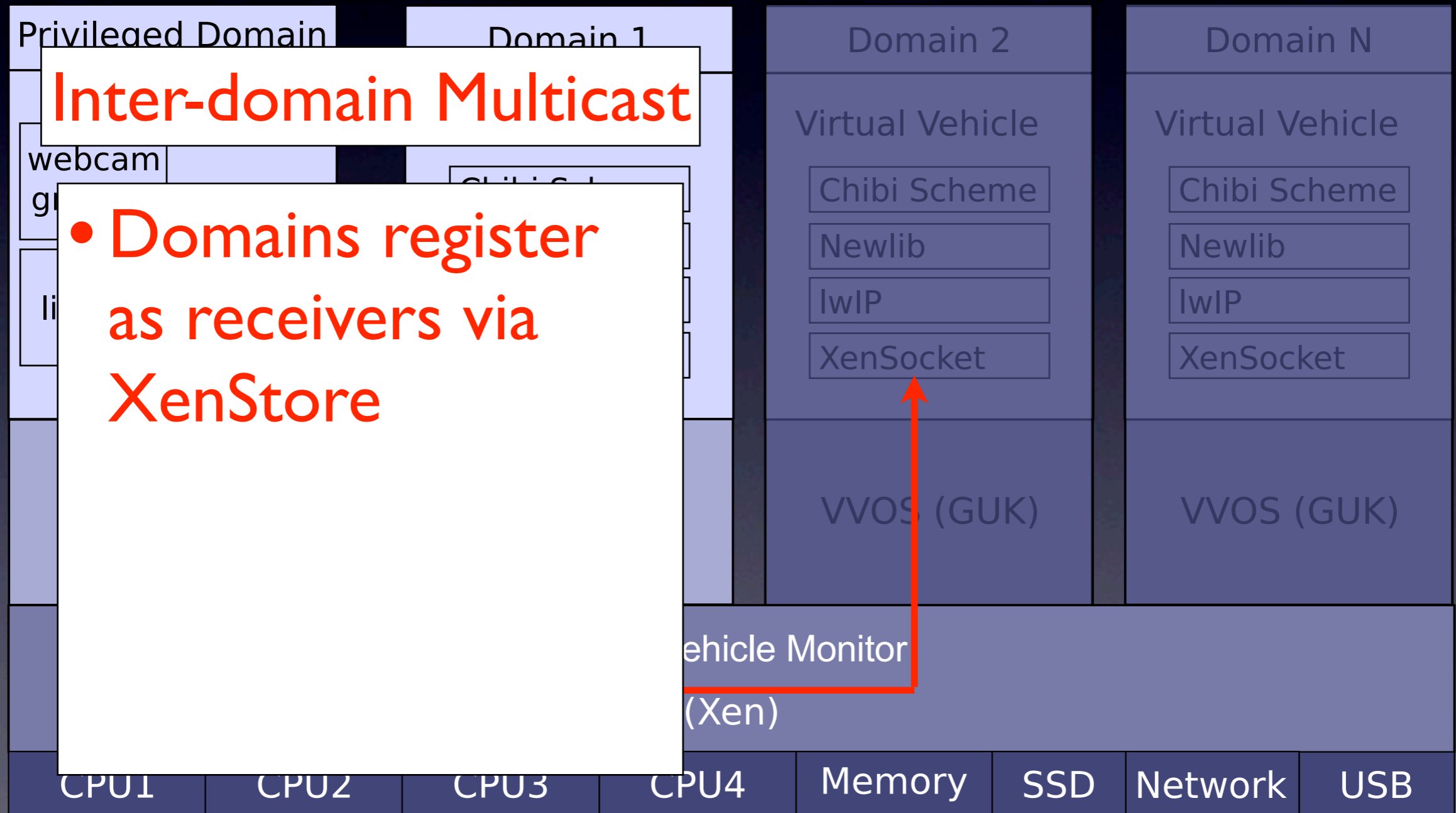
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



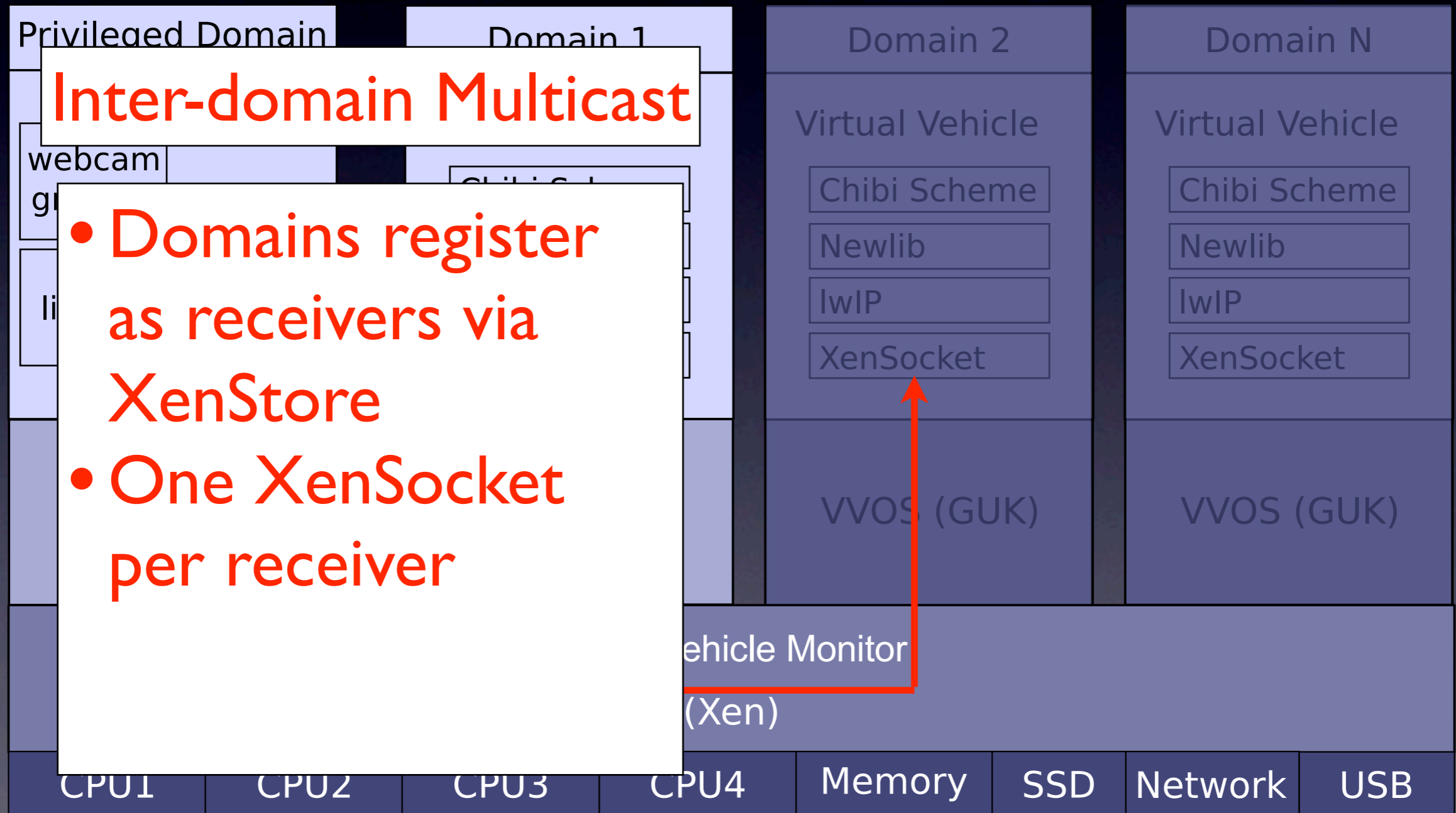
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



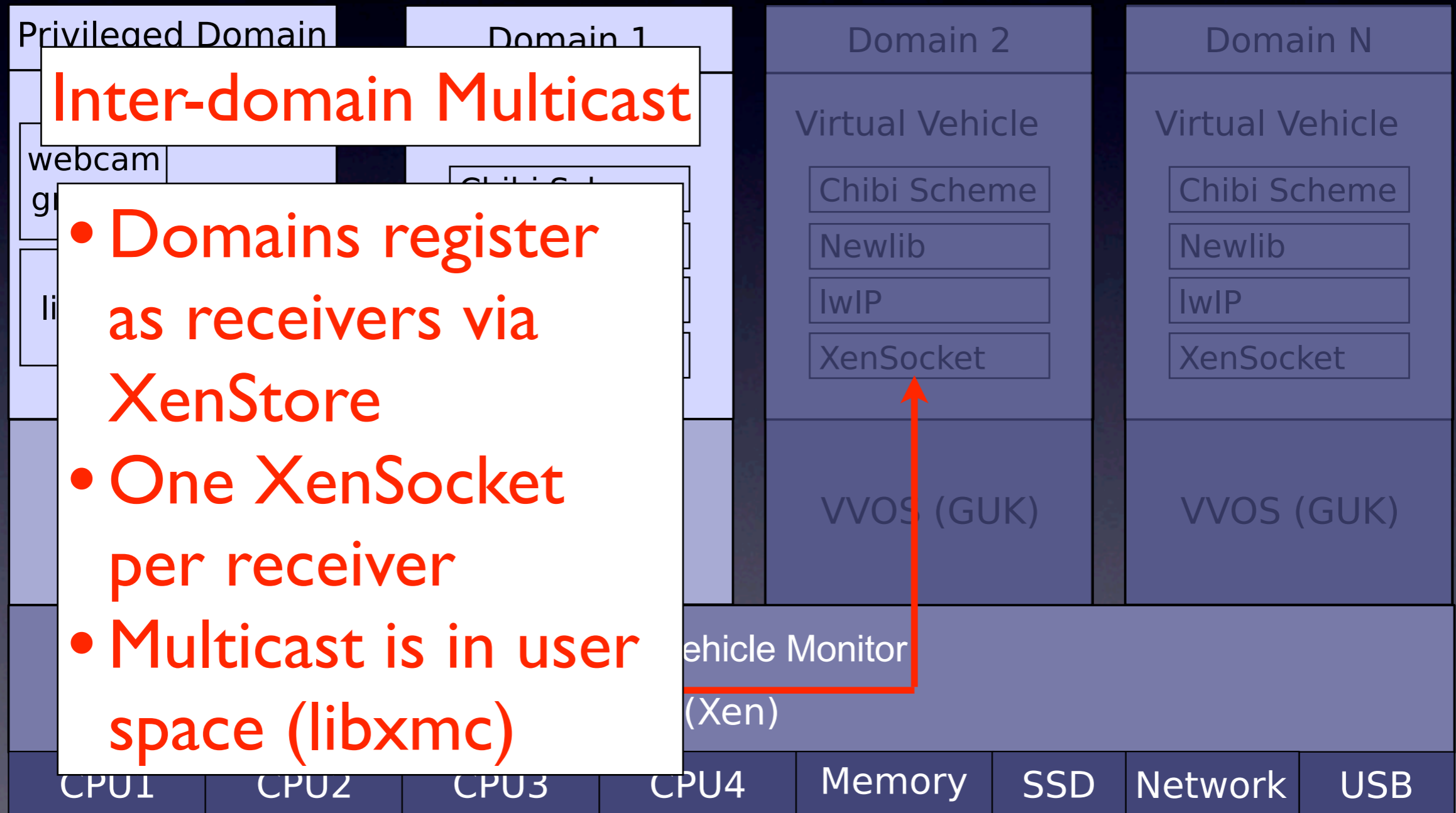
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



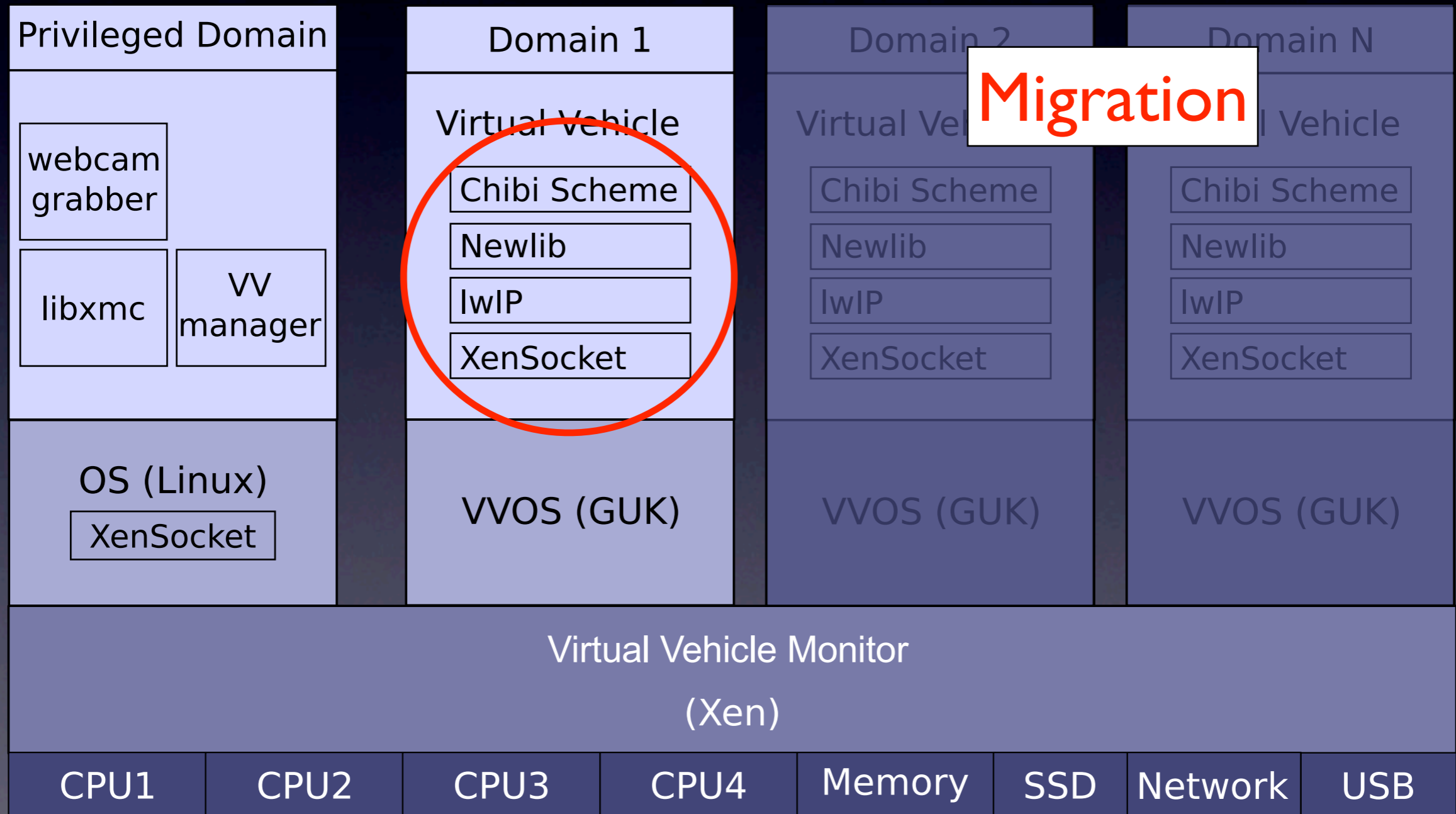
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



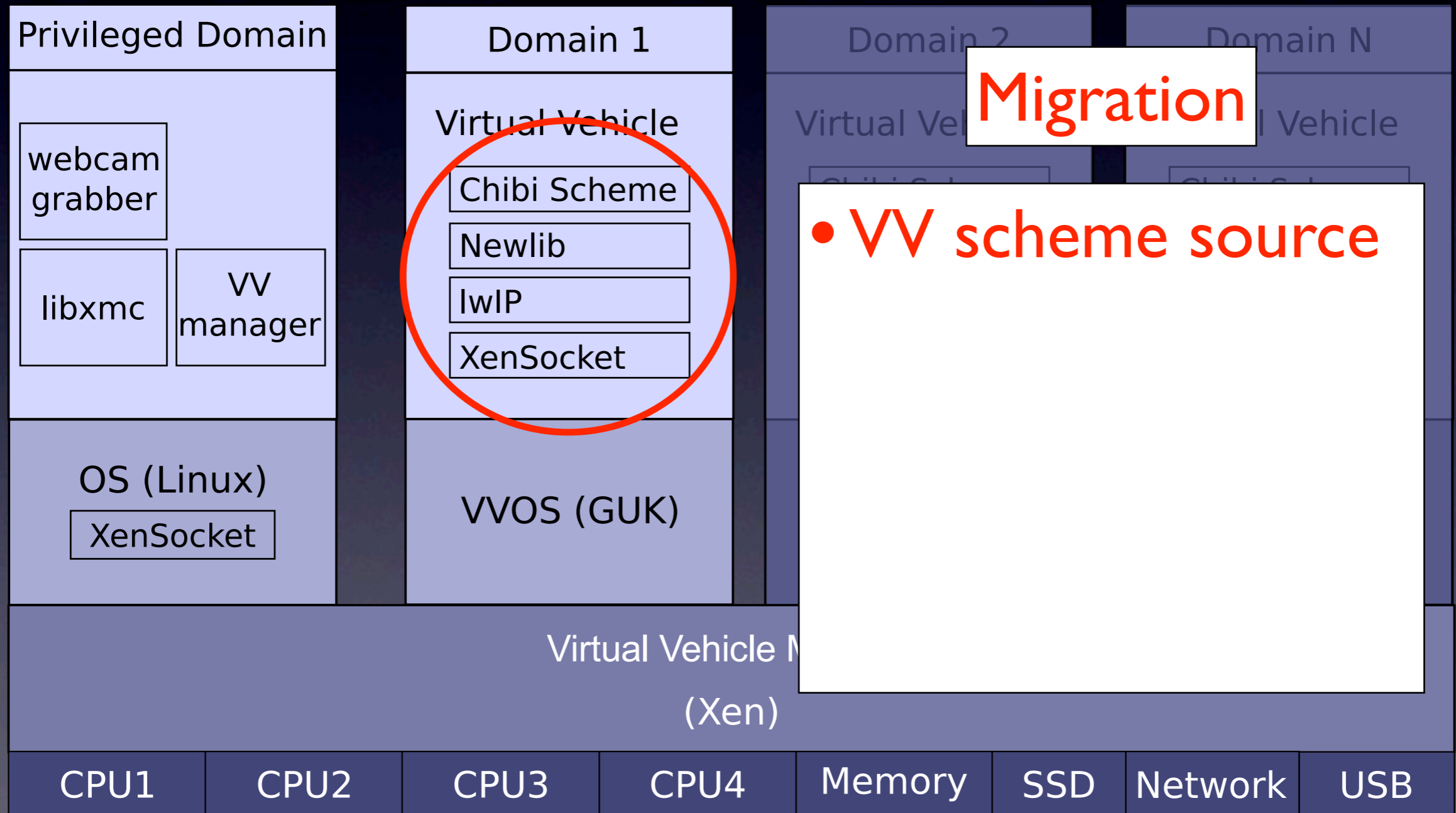
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



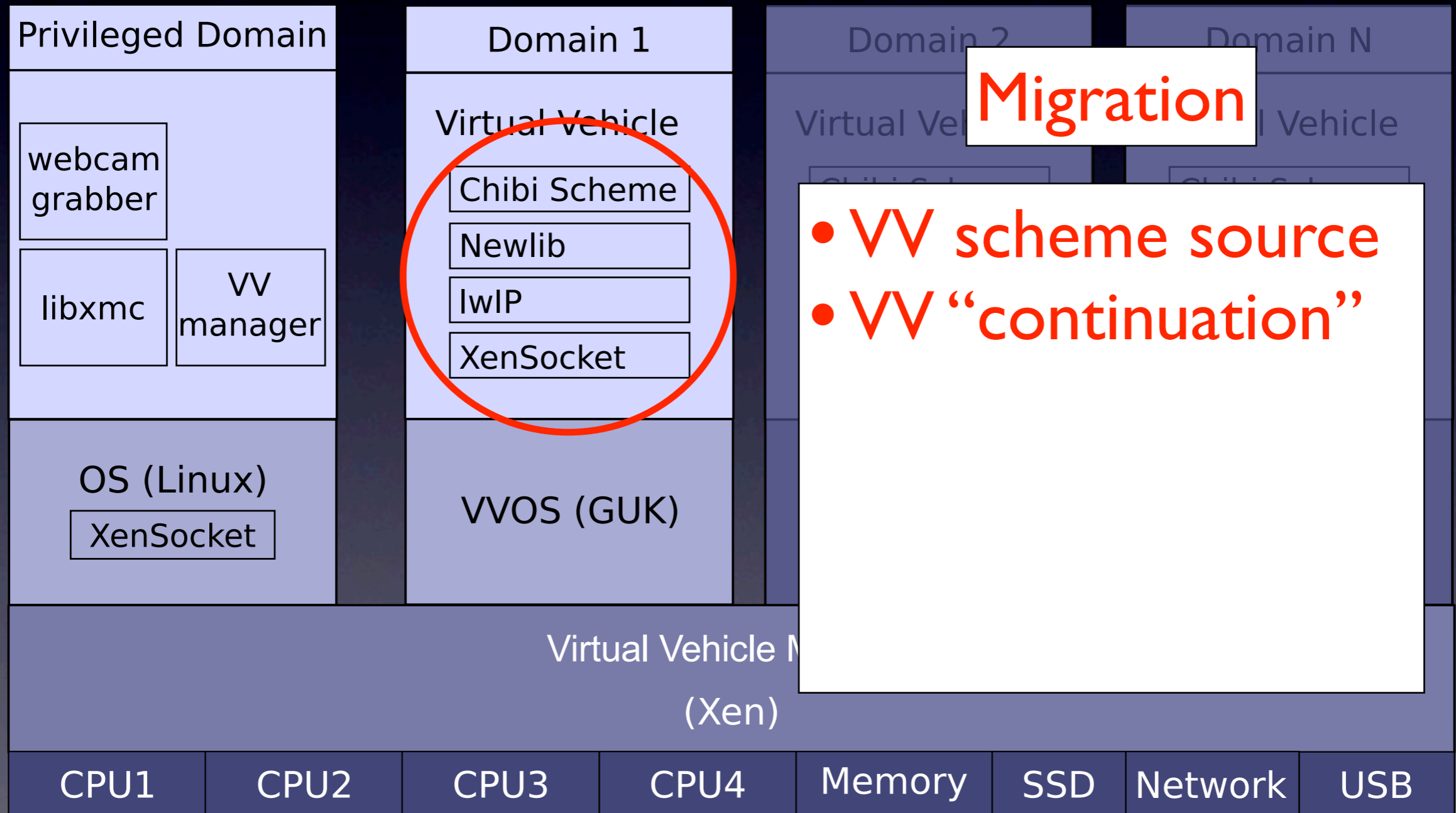
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



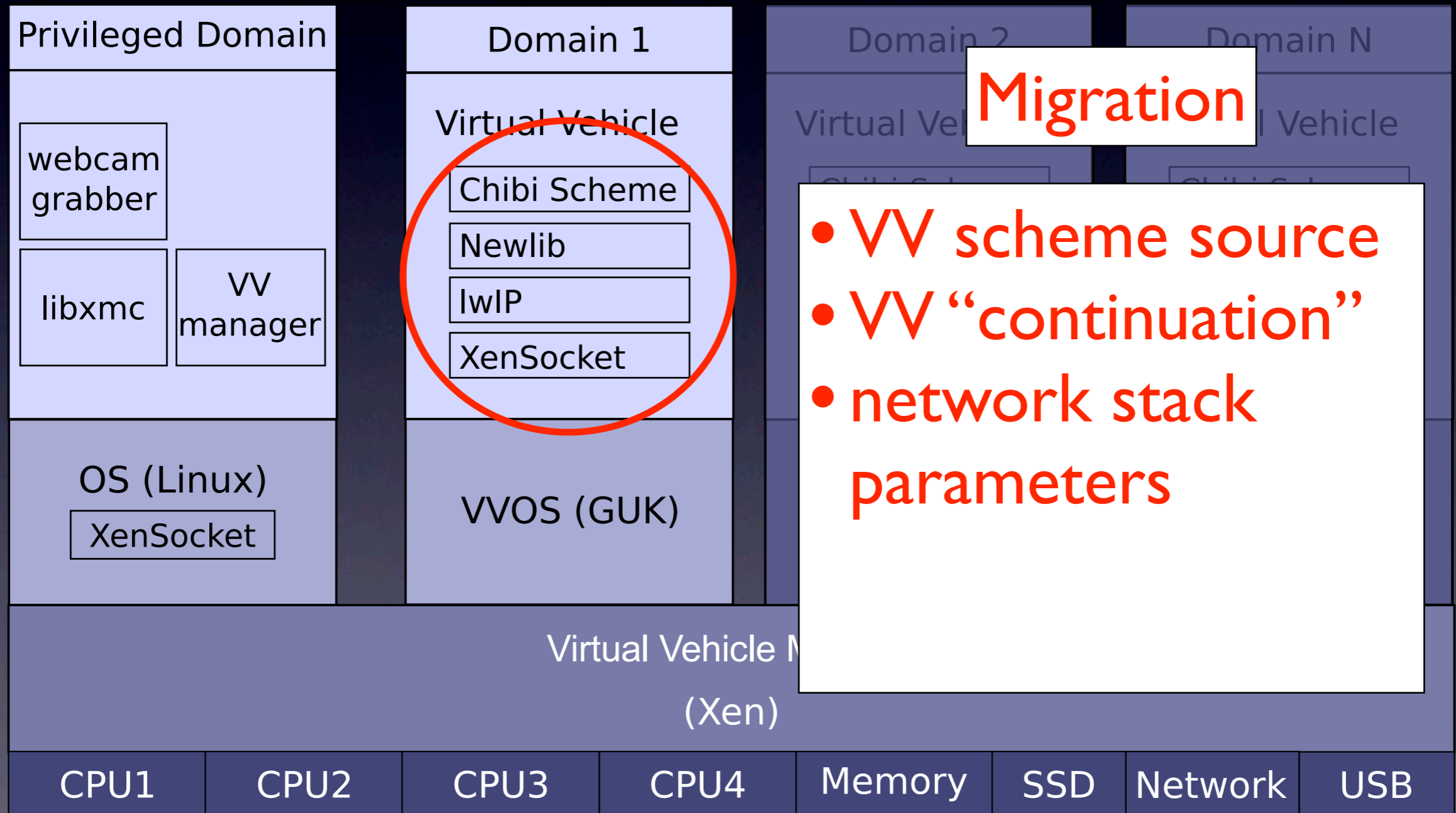
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



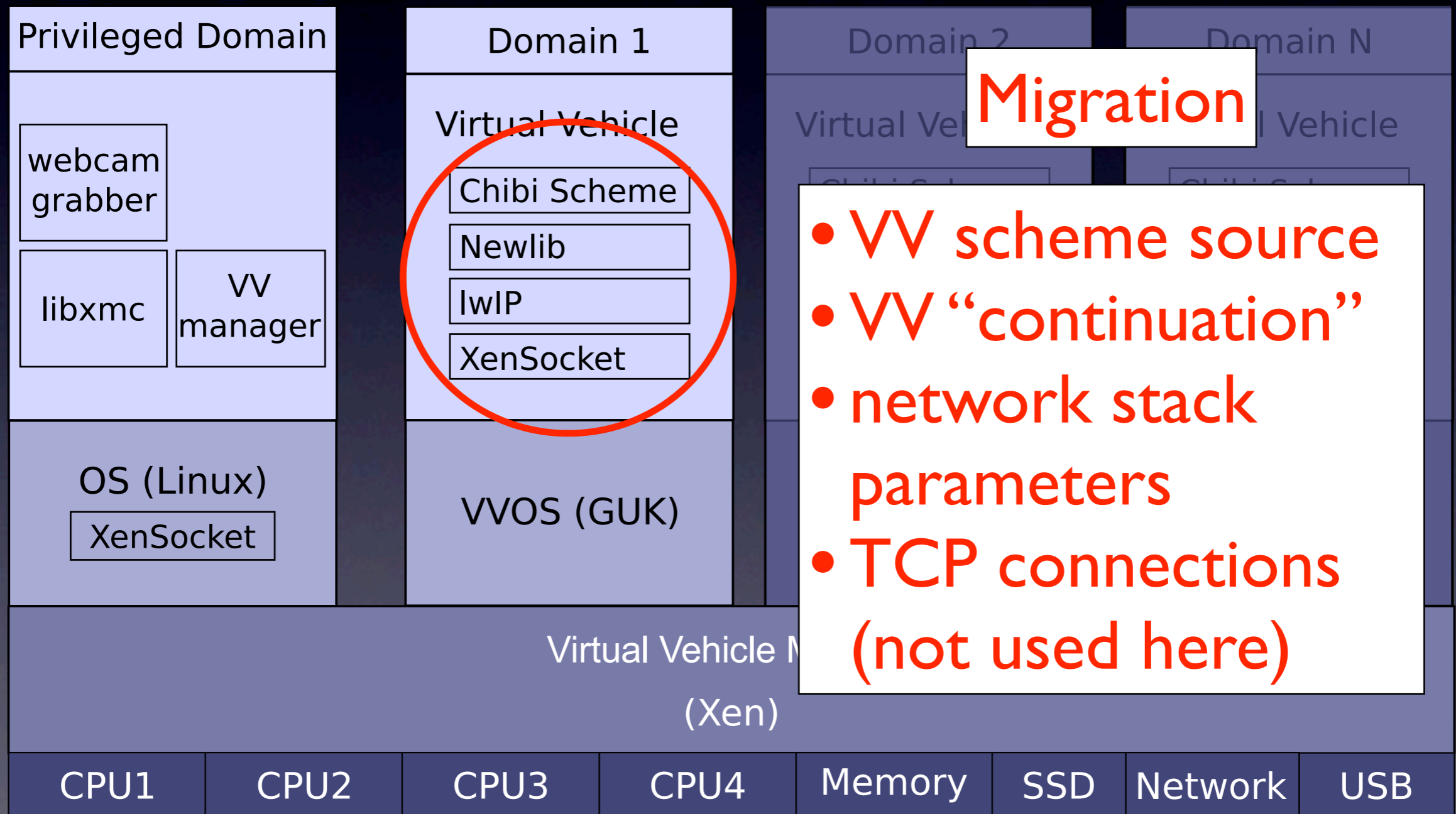
Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



Virtual Vehicle Monitor

(Xen, XenSockets, GUK, lwIP, Newlib, Chibi Scheme)



3 VVs on 2 Servers

3 VVs on 2 Servers

NAME	STATE	CPU(sec)	CPU(%)	MEM(k)	MEM(%)	MAXMEM(k)	MAXMEM(%)
Domain-0	-----r	86328	46.0	3741852	93.7	no limit	n/a
tramp-1	--b---	0	0.3	31744	0.8	32768	0.8
tramp-3	--b---	0	16.1	31744	0.8	32768	0.8
tramp-4	--b---	0	0.0	31744	0.8	32768	0.8
tramp-5	--b---	0	0.0	31744	0.8	32768	0.8
tramp-6	--b---	0	0.0	31744	0.8	32768	0.8
tramp-7	----p-	0	0.0	31744	0.8	32768	0.8

Delay Networks vds mem CPUs Repeat header Sort order Quit

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6773	root	20	0	52656	11m	3136	S	8	0.3	0:00.25	xm
4311	root	20	0	231m	18m	2252	S	5	0.5	46:40.89	xend
28568	root	20	0	117m	9.9m	4220	S	5	0.3	135:42.45	Xorg
4302	root	20	0	8628	1132	624	R	4	0.0	1:16.80	xenstored
1531	flandolt	20	0	31300	1848	1484	S	1	0.1	0:03.26	bouboule
28909	flandolt	20	0	19212	1480	1072	R	1	0.0	0:59.00	top

```
big-iron1$ sudo ./webFeed
Creating channel
-----src_fmt pixfmt:
RGB3
-----dst_fmt pixfmt:
MJPG
transferring data to: 2 domain(s)
```

```
big-iron3$ sudo ./webFeed
Creating channel
-----src_fmt pixfmt:
RGB3
-----dst_fmt pixfmt:
MJPG
transferring data to: 1 domain(s)
```

```
got line: Using config file "/tmp/trampd-cfg.sYrU2R".
got line: Started domain tramp-6 (id=199)
vm-pool: enqueued domain 199
Client 192.168.1.171:4098 accepted...
vm-pool: dequeued domain 196
Initiating state transfer with domain 196
vm-pool: creating new domain: name=tramp-7, ip=192.168.1.206
% xm create -p /tmp/trampd-cfg.5Fq0Q1
Waiting for domain 196 to become ready for state transfer
Copying state (3365 bytes) to domain 196...
Copying done.
Client 192.168.1.171:4098 done.
got line: main tool
got line: Using config file "/tmp/trampd-cfg.5Fq0Q1".
got line: Started domain tramp-7 (id=200)
vm-pool: enqueued domain 200
```

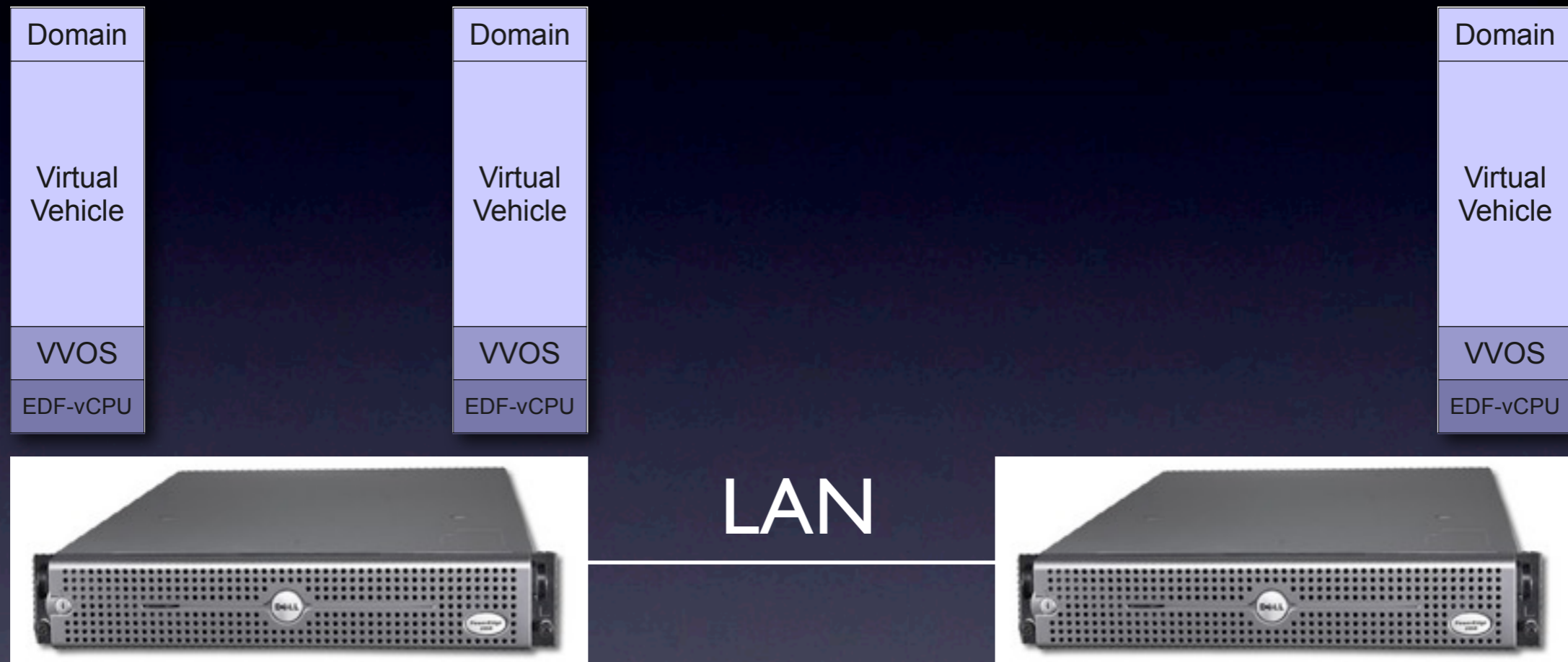
```
got line: Started domain tramp-4 (id=134)
vm-pool: enqueued domain 134
Client 127.0.0.1:38129 accepted...
vm-pool: dequeued domain 131
Initiating state transfer with domain 131
vm-pool: creating new domain: name=tramp-5, ip=192.168.1.204
% xm create -p /tmp/trampd-cfg.II4gIR
Waiting for domain 131 to become ready for state transfer
Copying state (3298 bytes) to domain 131...
Copying done.
Client 127.0.0.1:38129 done.
got line: main tool
got line: Using config file "/tmp/trampd-cfg.II4gIR".
got line: Started domain tramp-5 (id=135)
vm-pool: enqueued domain 135
Client 192.168.1.171:4097 accepted...
vm-pool: dequeued domain 132
Initiating state transfer with domain 132
vm-pool: creating new domain: name=tramp-6, ip=192.168.1.205
% xm create -p /tmp/trampd-cfg.MFJug0
Waiting for domain 132 to become ready for state transfer
Copying state (3440 bytes) to domain 132...
Copying done.
Client 192.168.1.171:4097 done.
got line: main tool
got line: Using config file "/tmp/trampd-cfg.MFJug0".
got line: Started domain tramp-6 (id=136)
vm-pool: enqueued domain 136
```

```
big-iron1$ ./tools/tramp-inject -i 192.168.1.171 --gw 192.168.1.1 --netmask 255.255.255.0 scheme-apps/demo.scm scheme-apps/config/demo/vc
hicle-01.scm
big-iron3$ ./tools/tramp-inject -i 192.168.1.171 --gw 192.168.1.1 --netmask 255.255.255.0 scheme-apps/demo.scm scheme-apps/config/demo/vc
hicle-02.scm
big-iron3$
```

```
big-iron3$ ./tools/tramp-inject -i 192.168.1.173 --gw 192.168.1.1 --netmask 255.255.255.0 scheme-apps/demo.scm scheme-apps/config/demo/vc
hicle-03.scm
big-iron3$
```

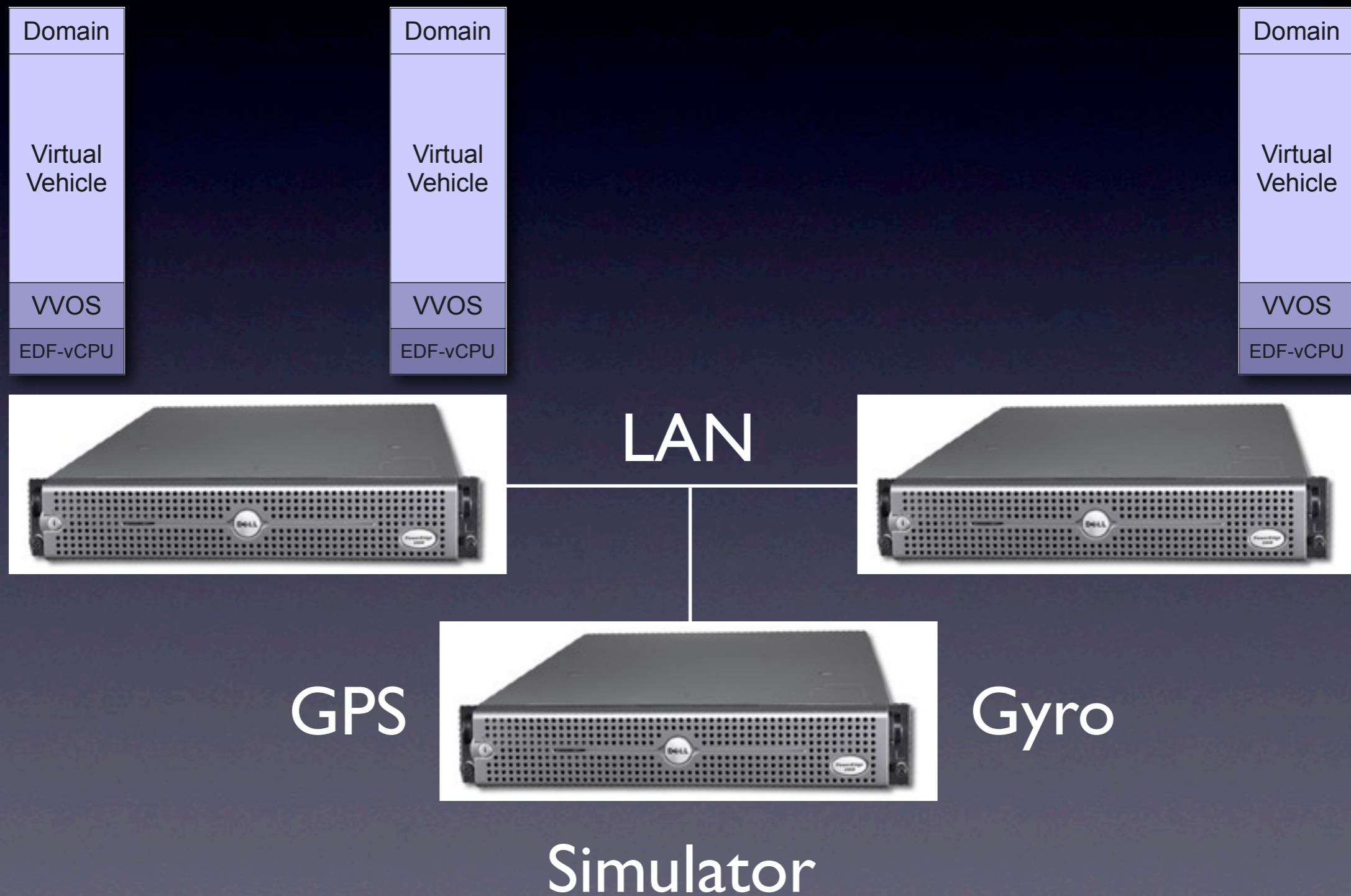

Future Work

(Simulating Motion, Flying Real Vehicles)



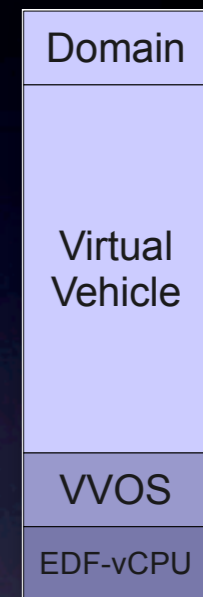
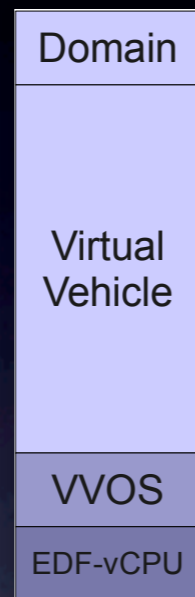
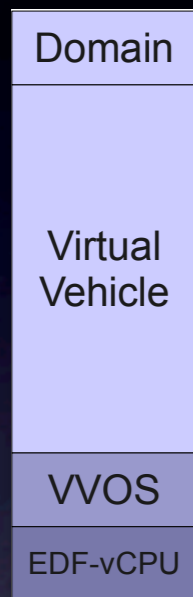
Future Work

(Simulating Motion, Flying Real Vehicles)



Future Work

(Simulating Motion, Flying Real Vehicles)

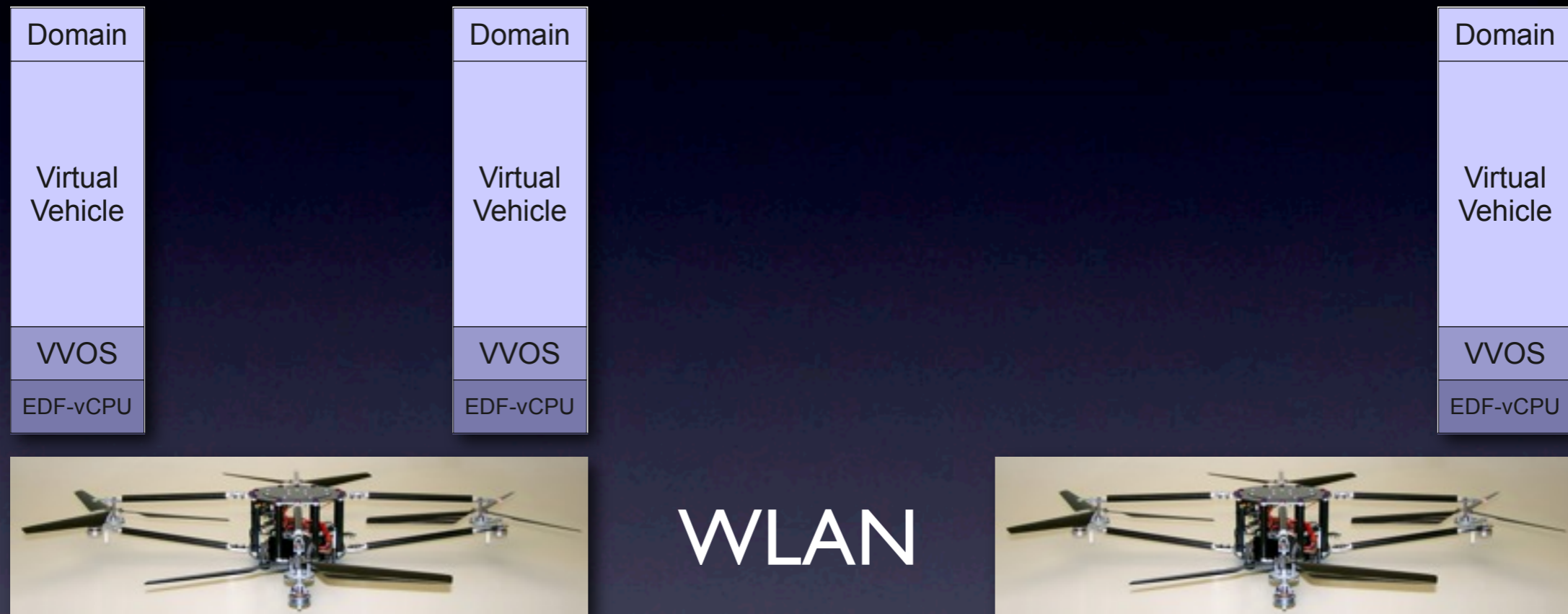


WLAN



Future Work

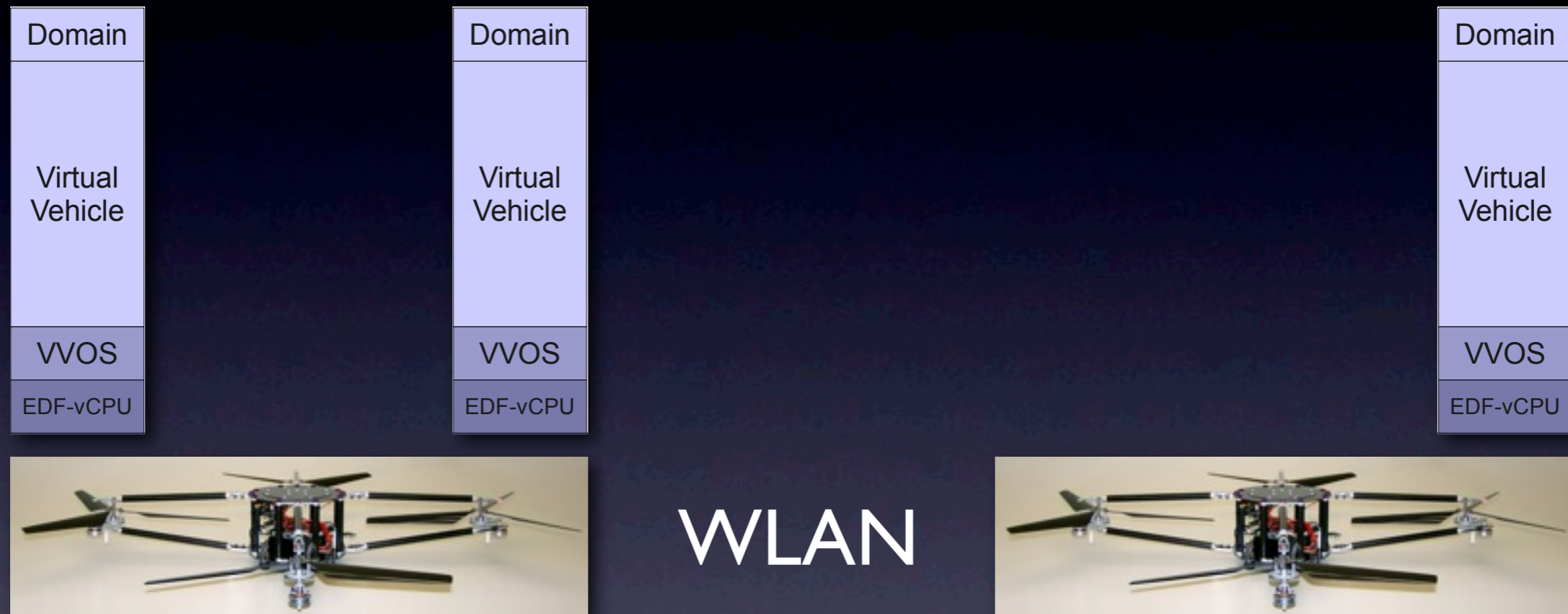
(Simulating Motion, Flying Real Vehicles)



- **read-only**, scheduled real vehicle flight plans:

Future Work

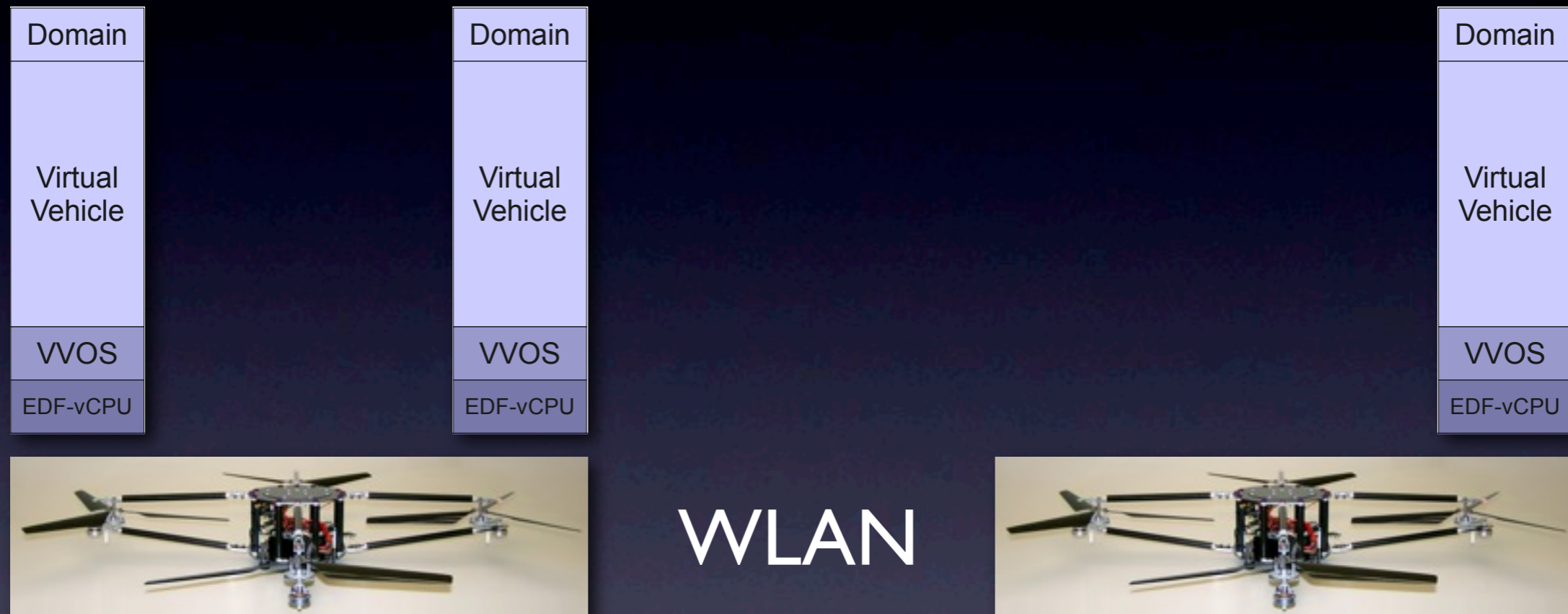
(Simulating Motion, Flying Real Vehicles)



- **read-only**, scheduled real vehicle flight plans:
 - ▶ **virtual-to-real** vehicle allocation problem

Future Work

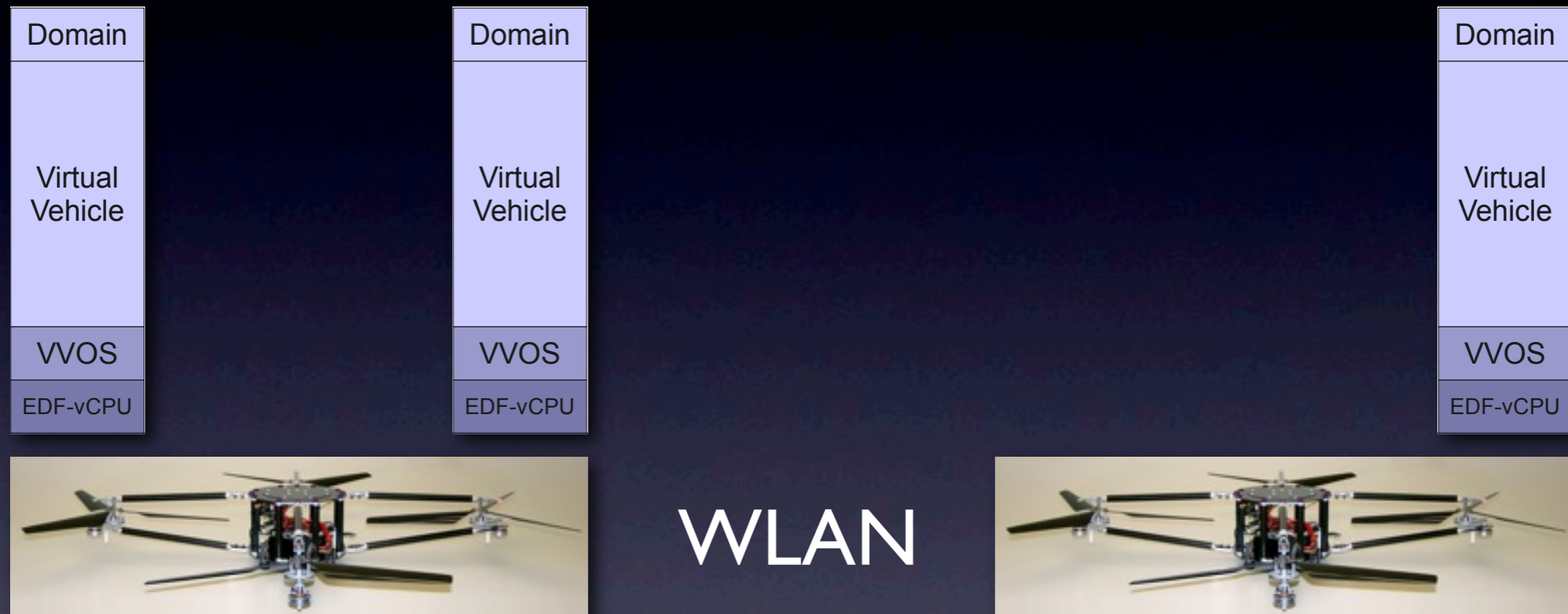
(Simulating Motion, Flying Real Vehicles)



- **read-only**, scheduled real vehicle flight plans:
 - ▶ **virtual-to-real** vehicle allocation problem
- **read-write**, on demand real vehicle flight plans:

Future Work

(Simulating Motion, Flying Real Vehicles)



- **read-only**, scheduled real vehicle flight plans:
 - ▶ **virtual-to-real** vehicle allocation problem
- **read-write**, on demand real vehicle flight plans:
 - ▶ **real-to-virtual** vehicle allocation problem



Thank you

Check out:
eurosys2011.cs.uni-salzburg.at