

OpenSIPS 2.0 ***a programmable SIP framework***

Bogdan-Andrei Iancu
CEO Voice System
Founder OpenSIPS Project

Once upon a time there was a powerful and flexible SIP Server

....there was OpenSIPS doing tens of thousands CPS.

BUT....

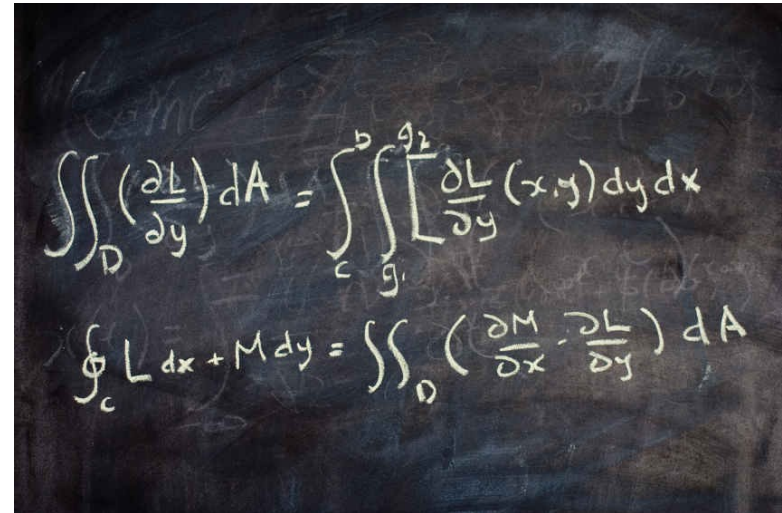
SIP Low level awareness

- **you still need to be aware of and handle low level SIP bits and pieces (transactions, dialogs, NAT, etc) to make it work**
- **you cannot focus only on service creation**



Configuration skills

- you are required to learn the custom OpenSIPS scripting language
- you are limited to what OpenSIPS script language has to offer
- the script language is not integration friendly



The image shows two mathematical equations written on a chalkboard. The first equation is a double integral over a domain D of the partial derivative of L with respect to y, multiplied by dA, equal to a double integral over a region bounded by curves g1 and g2 from x=c to x=b of the partial derivative of L(x,y) with respect to y, multiplied by dy dx. The second equation is a line integral over a closed curve C of L dx + M dy, equal to a double integral over a domain D of the partial derivative of M with respect to x minus the partial derivative of L with respect to y, multiplied by dA.

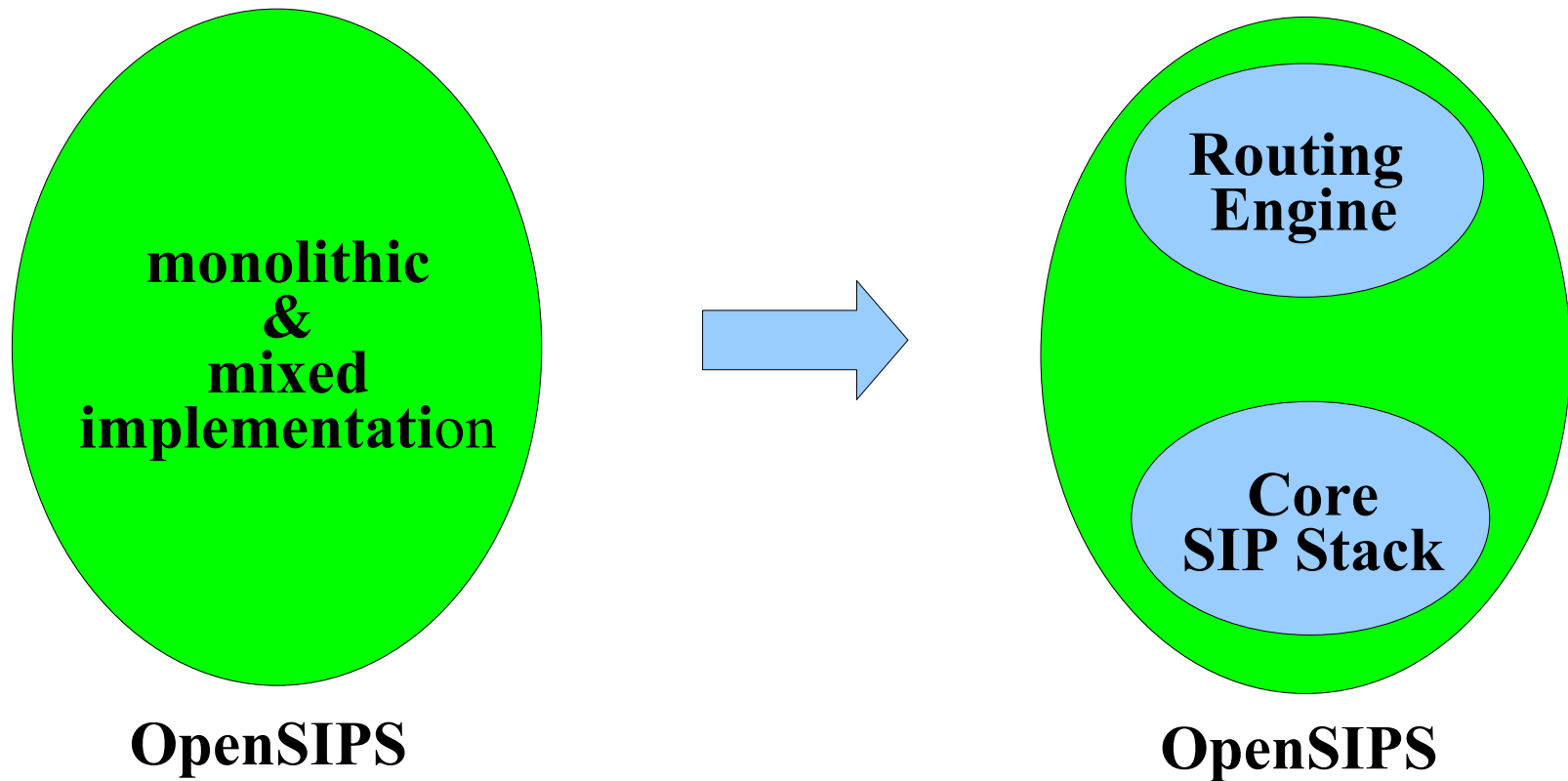
$$\iint_D \left(\frac{\partial L}{\partial y} \right) dA = \int_c^b \int_{g_1}^{g_2} \left[\frac{\partial L}{\partial y} (x,y) \right] dy dx$$
$$\oint_C L dx + M dy = \iint_D \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} \right) dA$$

Horizontal Scalability

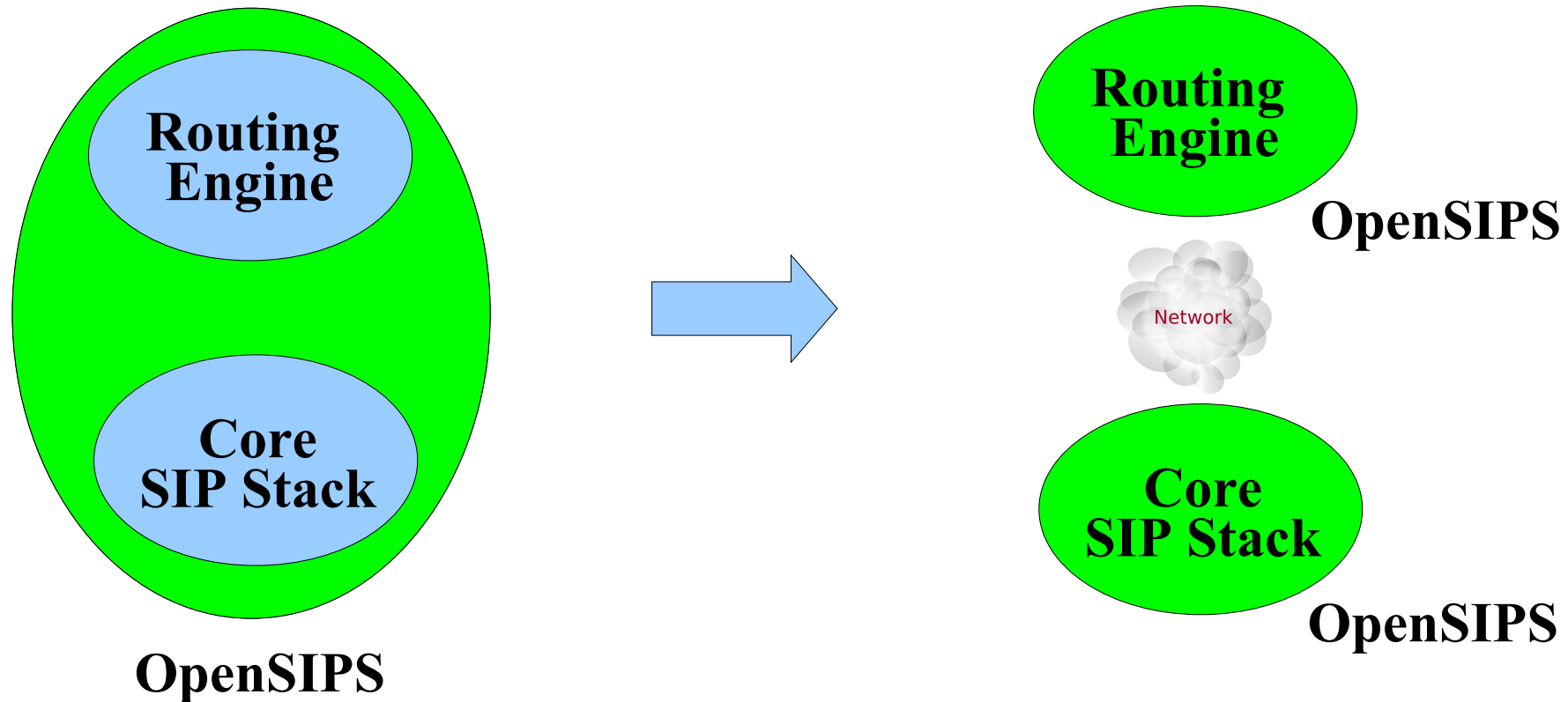
- cannot scale with a single instance, no matter how powerful it is
- clustering must be naturally achieved
- traffic and data sharing across all nodes in cluster



OpenSIPS 2.0 == 42

STEP 1 – separation of SIP stack and routing logic

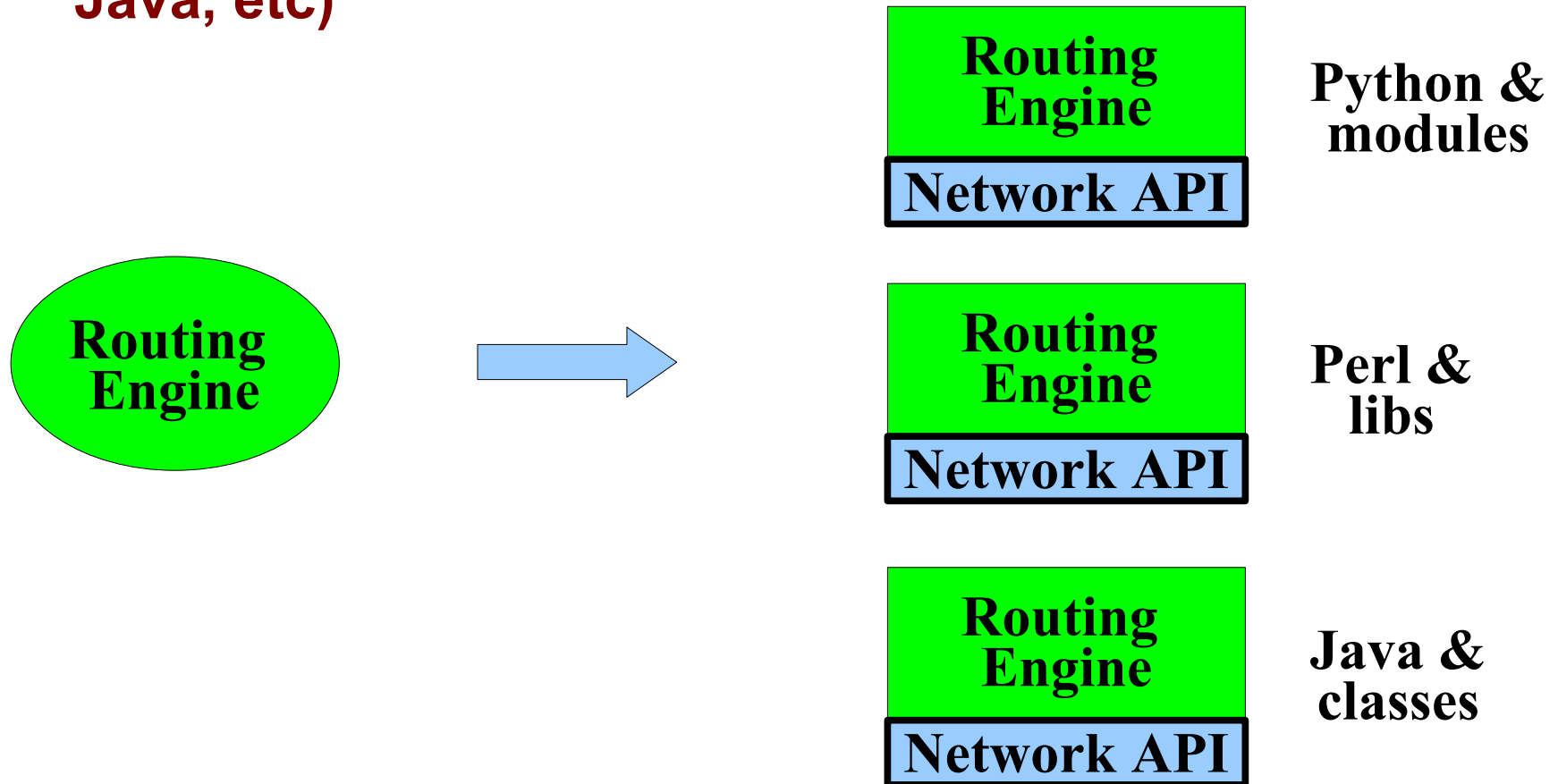
STEP 2 – decouple routing & core to get separated & independent applications



Solved:

- no need to deal SIP low level, just to control and interact with it
- you can focus on service creation without taking care of SIP specific details
- achieve vertical scalability (routing logic and core may be on different machines)
- optimize the processing – the Core part (SIP stack) may automatically and transparent handle certain SIP events (like CANCEL, ACK, etc)

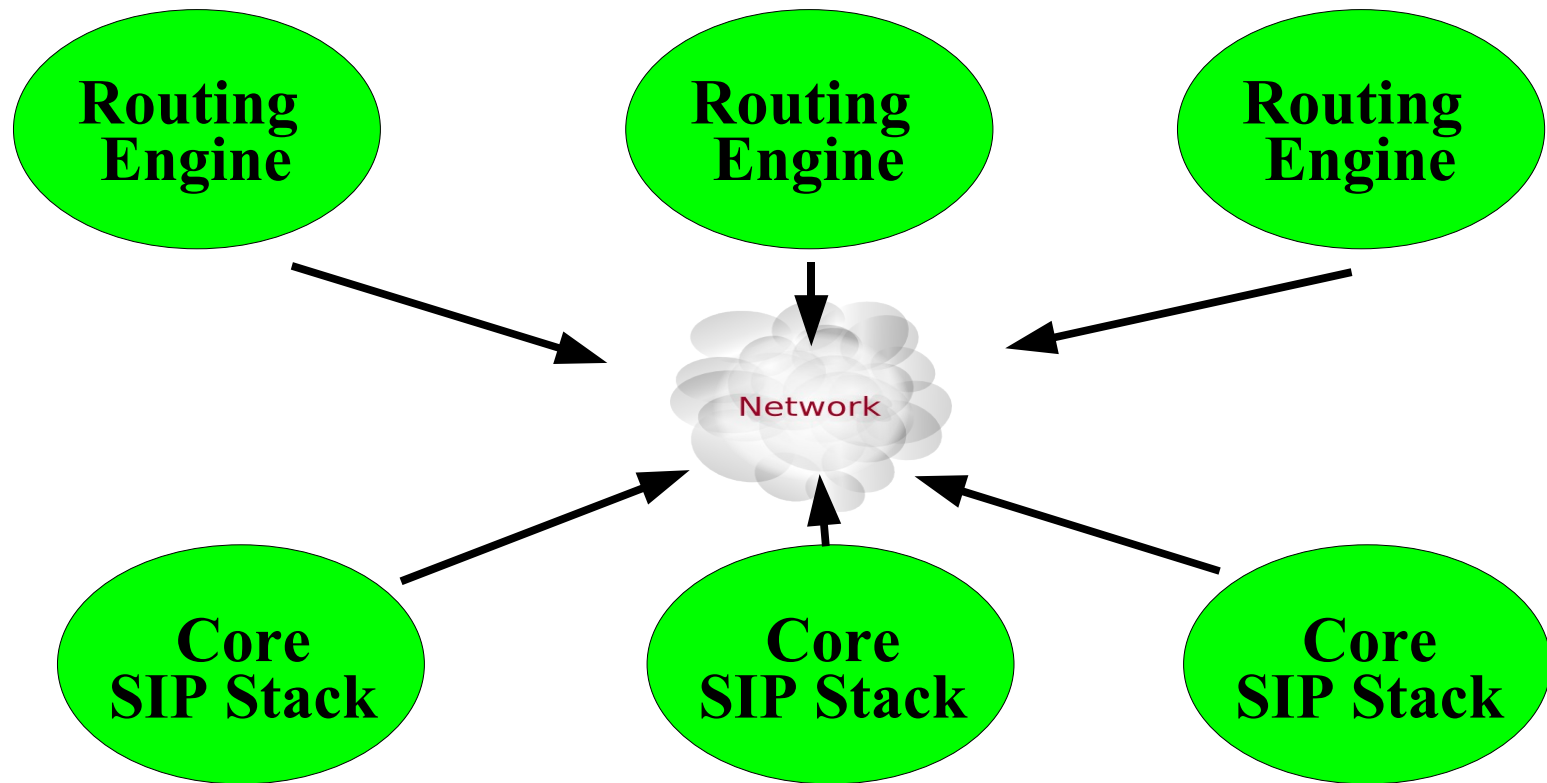
STEP 3 – programmable routing logic (Perl, Python, Java, etc)



Solved:

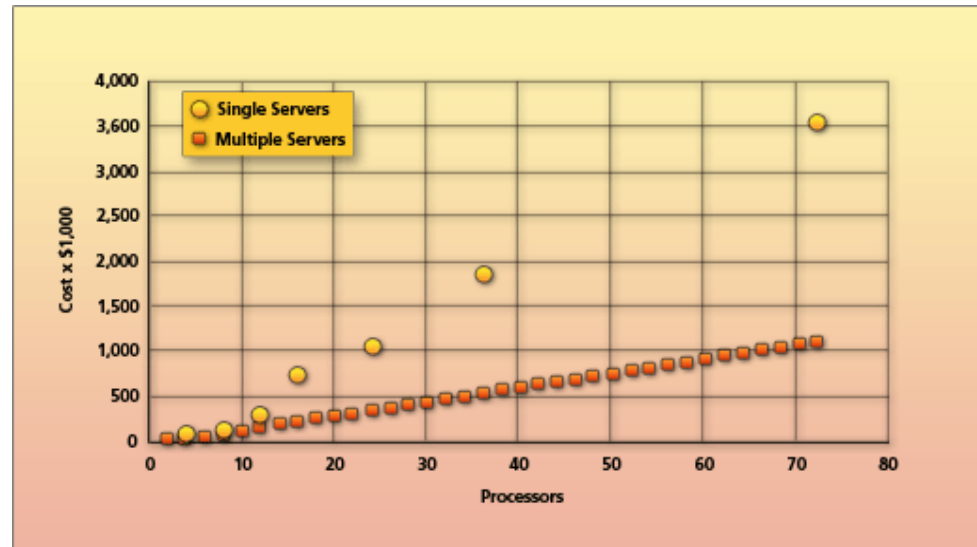
- no more custom language for scripting – you can use your own favorite language (any)
- scripting is no more limited – you can take full advantages of the capabilities (as scripting) and already existing functionality from the high-level programming languages
- integration (with whatever other apps in whatever other languages) become trivial as what language is used is no more a limitation, but rather an advantage
- routing logic can be actually part of other larger application

STEP 4 – horizontal scalability for both Core and Logic



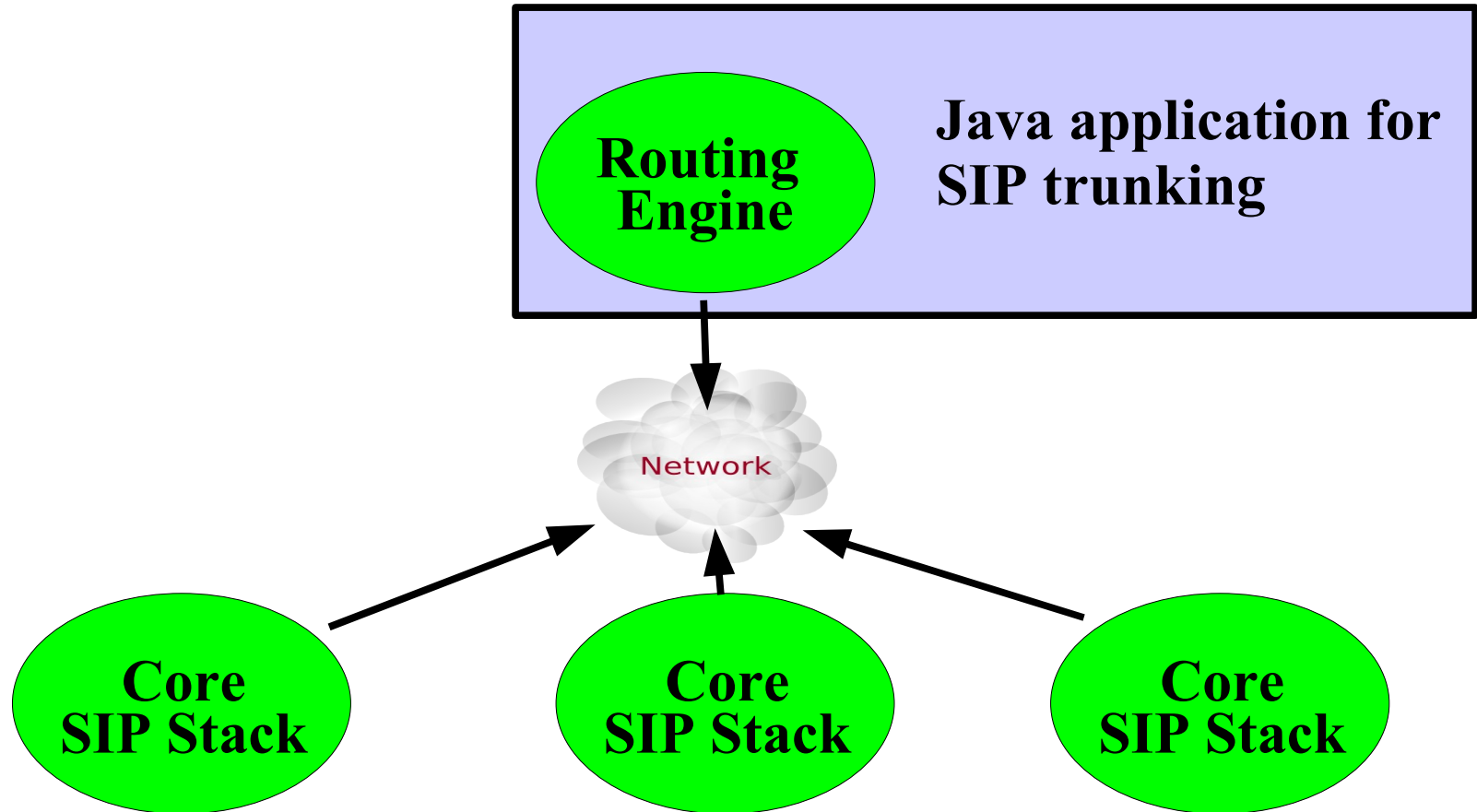
Solved:

- horizontal scalability – each part (core and logic) may individually scale across several machines.
- the logic will be responsible for clustering (service and data) by providing to Core part data storage support
- it is cheaper to scale (for same number of CPUs) with several machines, rather than only one

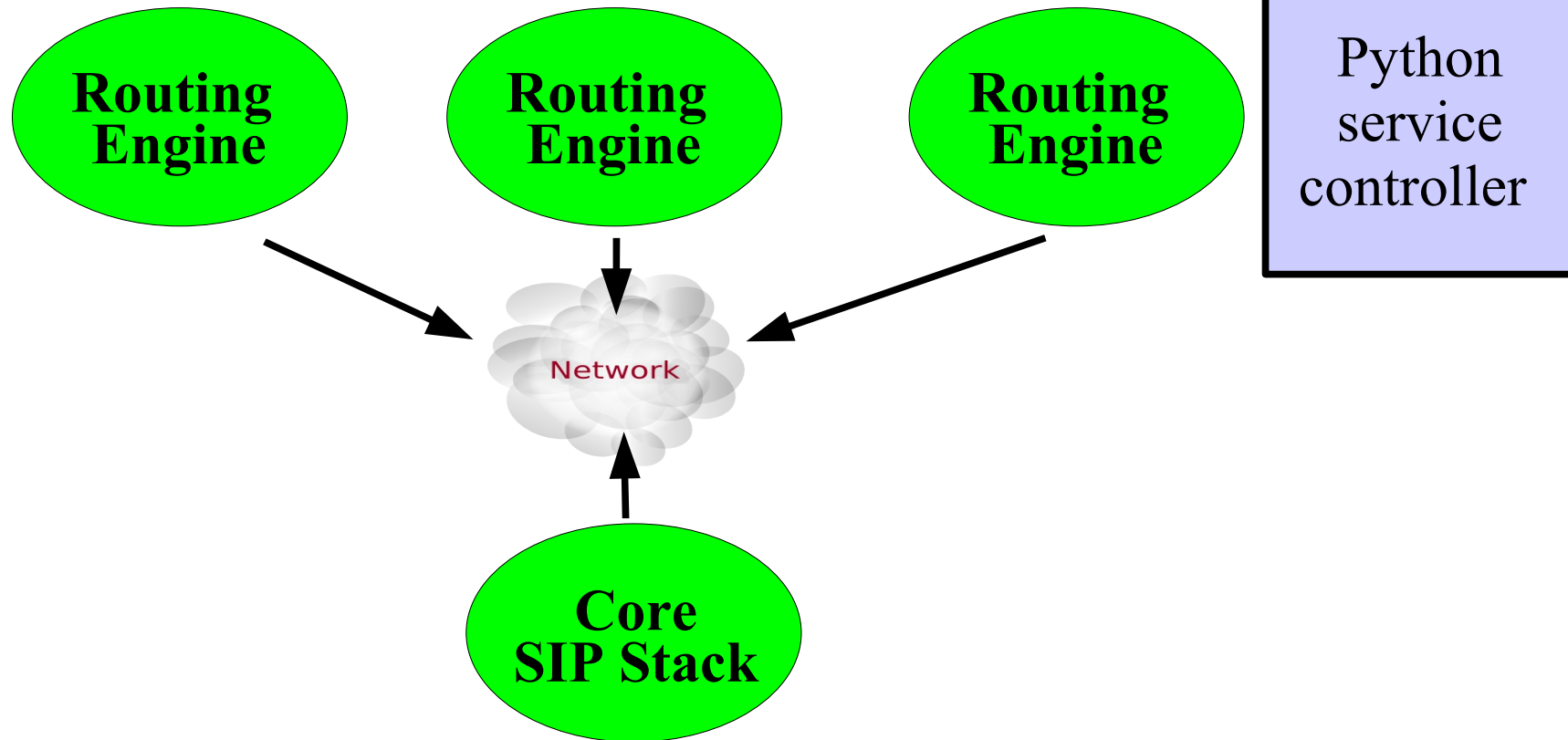


Examples

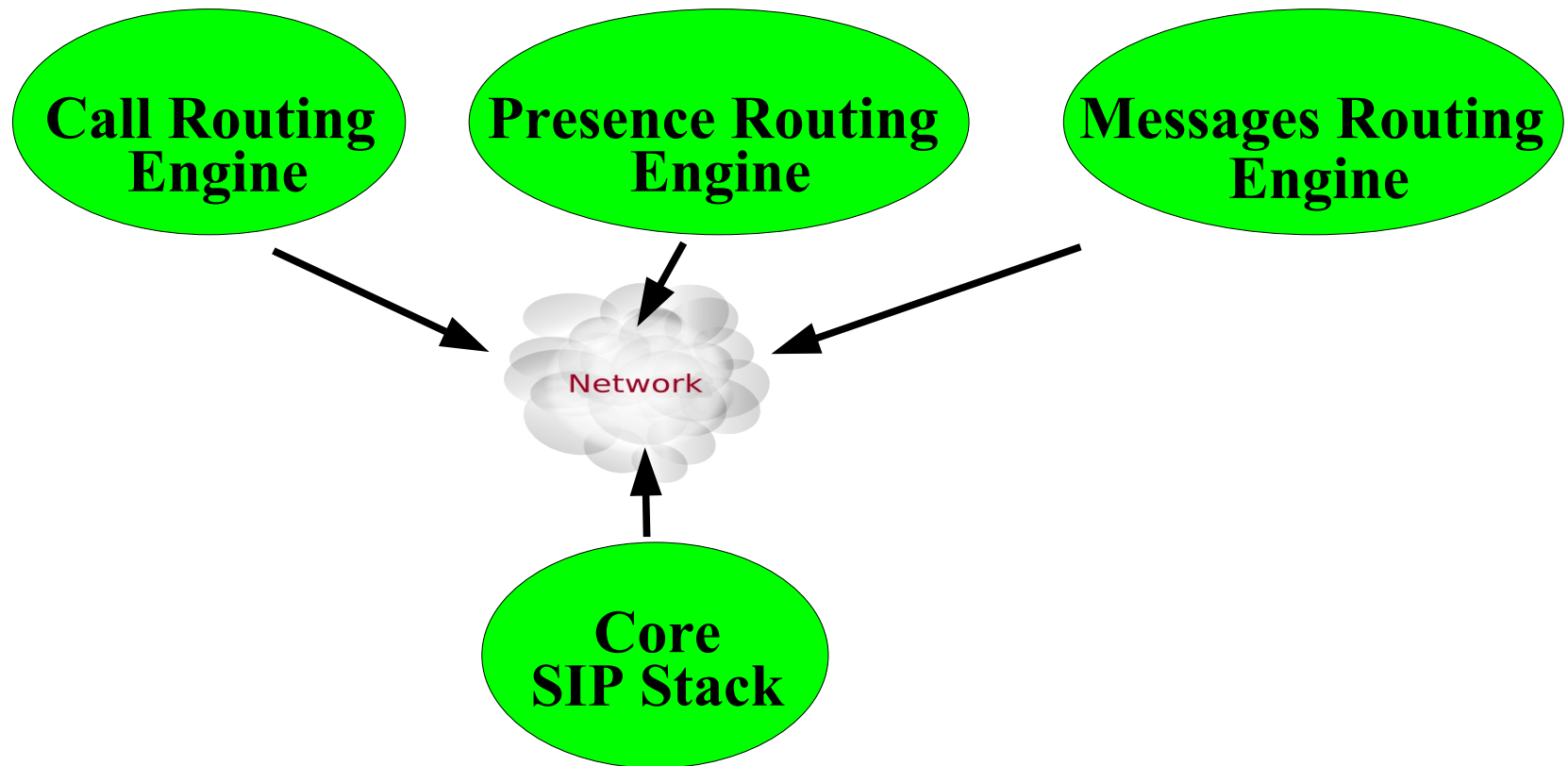
Intensive traffic, simple logic



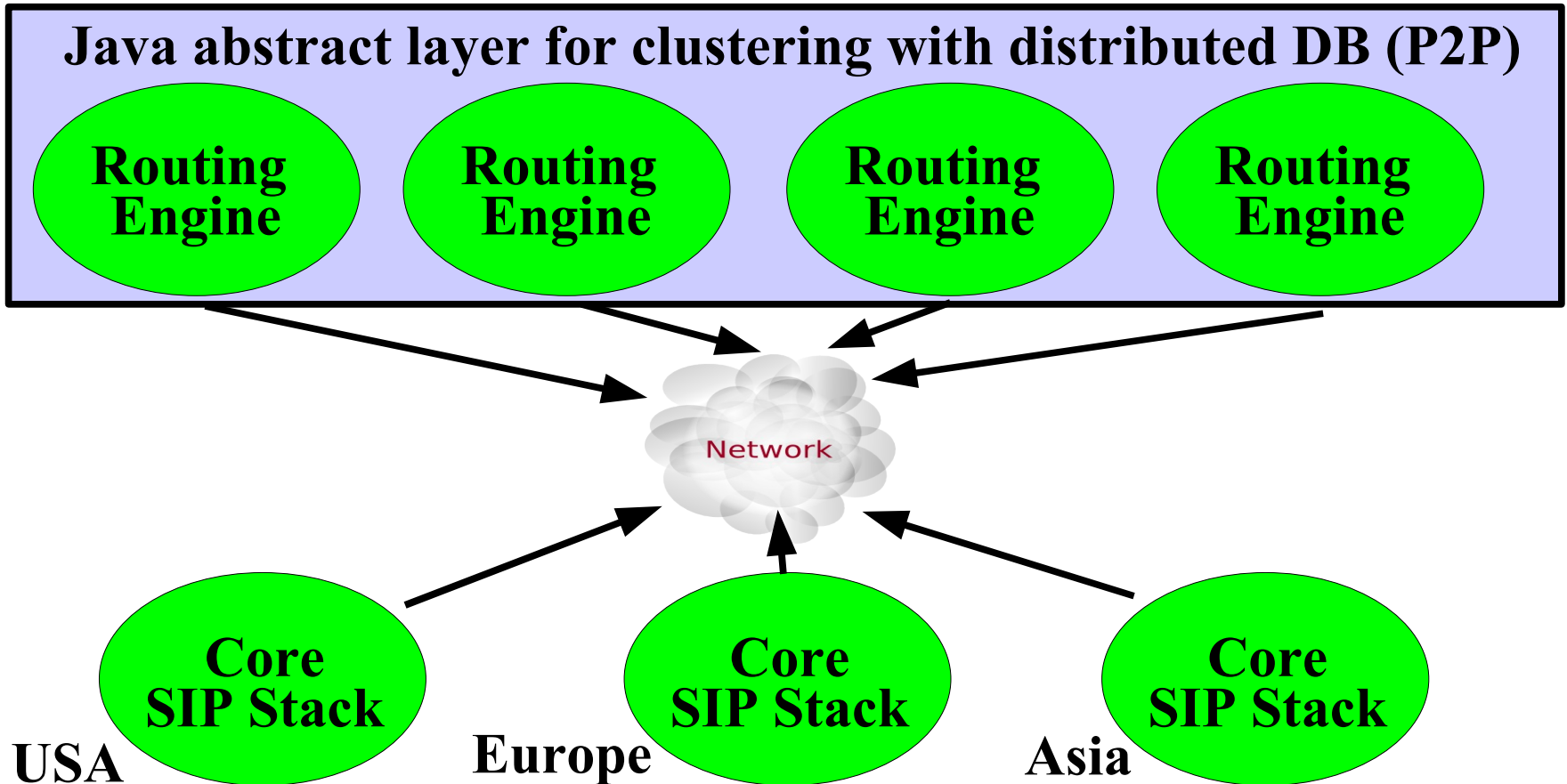
Heavy logic and integration



Specialized logics



Geographical Clustering



Thank you for your attention
You can find out more at www.opensips.org
bogdan.iancu@voice-system.ro

Questions are welcome