Transformations of Attributed Graphs with Cloning

Dominique Duval
with Rachid Echahed, Frédéric Prost and Leila Ribeiro

LJK-LIG, Université de Grenoble, France
Universidade Federal do Rio Grande do Sul, Brazil

June 26., 2014 – LJK – Grenoble – Bipop-Casys seminar

- Rewriting / Transformation of:
  - terms
  - graphs
  - attributed graphs

- with algebraic methods:
  - PO (add, merge)
  - DPO (add, merge, delete)
  - SqPO (add, merge, delete, copy)

Outline

Term rewriting

Graph transformation

Attributed graph transformation: FASE 2014

Example: term rewriting

$$1 + 1 = 2\,?$$

Example: term rewriting

$$1 + 1 \rightsquigarrow 2\,?$$

Example: term rewriting

$$1 + 1 \rightsquigarrow 2\,?$$

Specification:

- N
- $0 : N$, $s : N \to N$, $+ : N, N \to N$

Rules:

$(R_0) \colon x + 0 \rightsquigarrow x \qquad (R_1) \colon x + s(y) \rightsquigarrow s(x + y)$

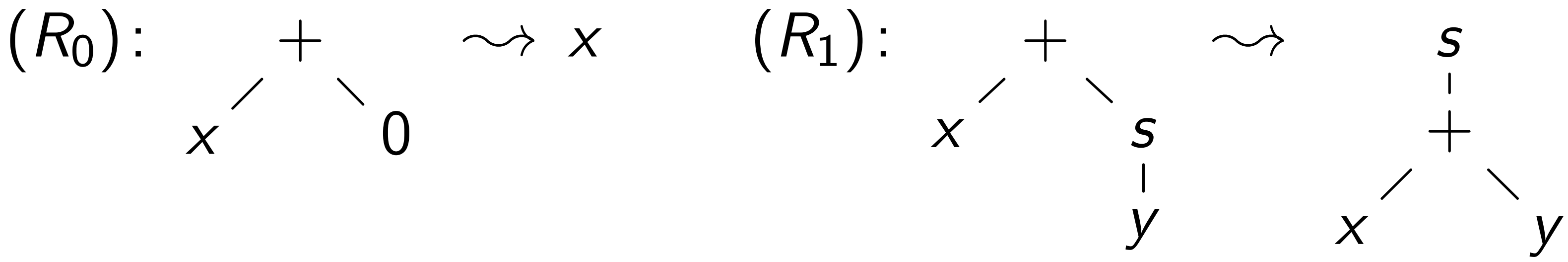
Reduction:

$$s(0) + s(0) \overset{(R_1)}{\rightsquigarrow} s(s(0) + 0) \overset{(R_0)}{\rightsquigarrow} s(s(0))$$

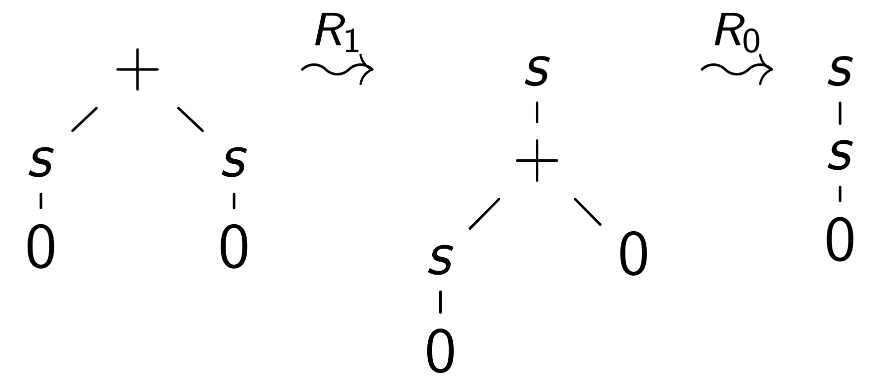
Example: term rewriting

$$1 + 1 \rightsquigarrow 2\,?$$

Rules (dim. 2):



Reduction (dim. 2):



Terms and graphs

A term "is" a tree, and a tree "is" a graph.

However:

- Trees are defined inductively ("generalized" lists):

$$T ::= r \mid TrT$$

- Graphs are defined as "presheaves" ("generalized" sets):

$$\begin{tikzcd} V & E \arrow[l, bend right, "s"'] \arrow[l, bend left, "t"] \end{tikzcd}$$

Consequence:
It is difficult to adapt term rewriting to graphs!

Outline

Term rewriting

Graph transformation

Attributed graph transformation: FASE 2014

Graph transformation

L, R, G, H are graphs.

Given a rewrite rule:

$$L \rightsquigarrow R$$

and a matching:

$$\begin{tikzcd} L \arrow[d, "\subseteq"] \\ G \end{tikzcd}$$

a rewrite step builds H by replacing
the occurrence of L in G
by some occurrence of R in H:

$$\begin{tikzcd} R \arrow[d, "\subseteq"] \\ H \end{tikzcd}$$

Graph transformation

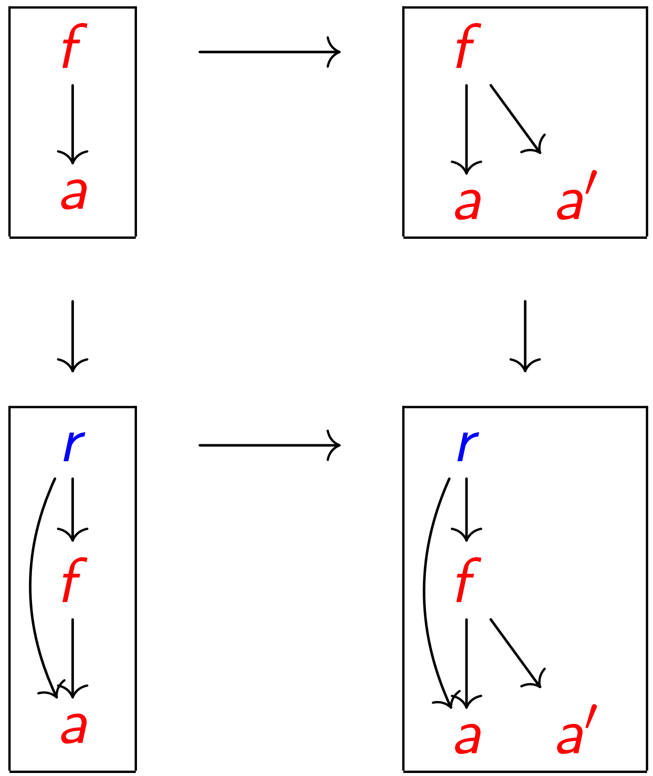
A rewrite step:

$$\begin{tikzcd} \boxed{L} \arrow[r, rightsquigarrow] \arrow[d] & \boxed{R} \arrow[d] \\ \boxed{G} \arrow[r, rightsquigarrow] & \boxed{H} \end{tikzcd}$$

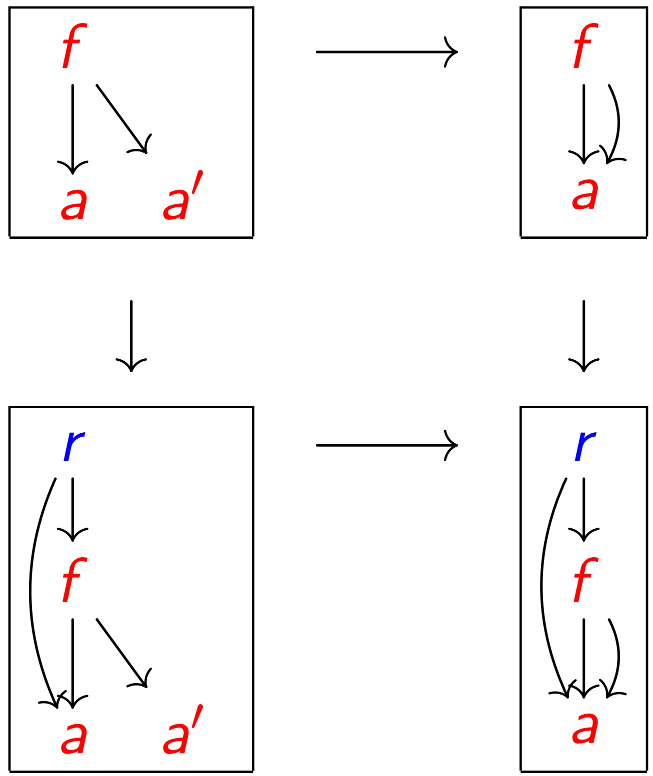
Elementary transformations:

- ADD
- MERGE
- DELETE
- COPY (= clone)

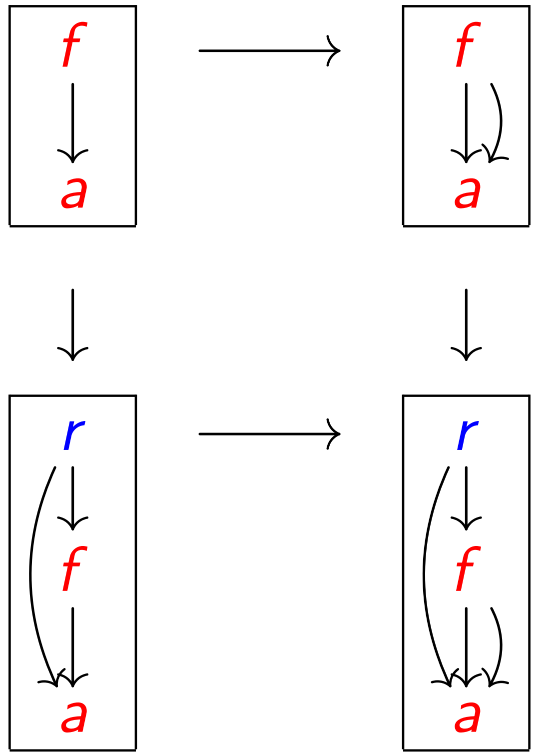
Graph transformation: ADD



Graph transformation: MERGE



Graph transformation: ADD and MERGE



Graph transformation: ADD and MERGE

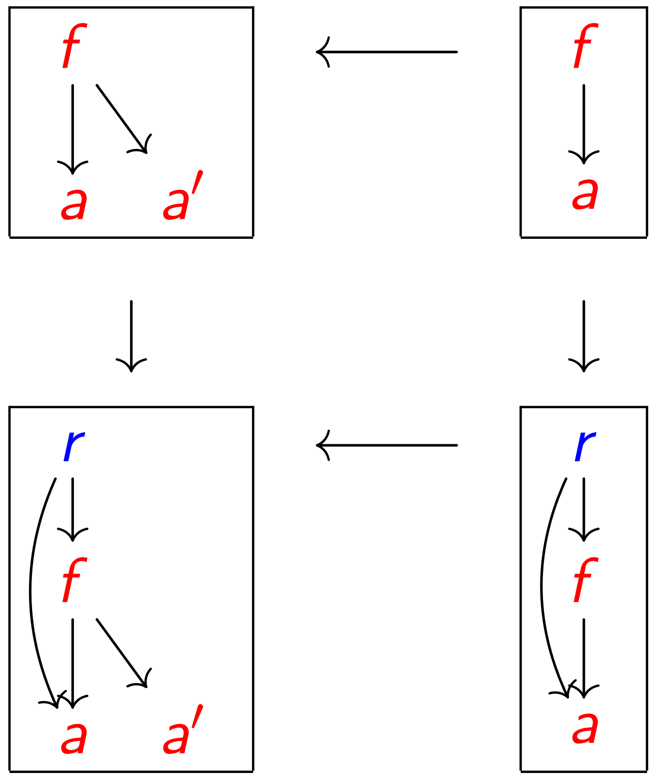
- A rule $L \rightsquigarrow R$ is a graph homomorphism $L \to R$ from L to R
- A step

$$\begin{tikzcd} \boxed{L} \arrow[r, rightsquigarrow] \arrow[d] & \boxed{R} \arrow[d] \\ \boxed{G} \arrow[r, rightsquigarrow] & \boxed{H} \end{tikzcd}$$

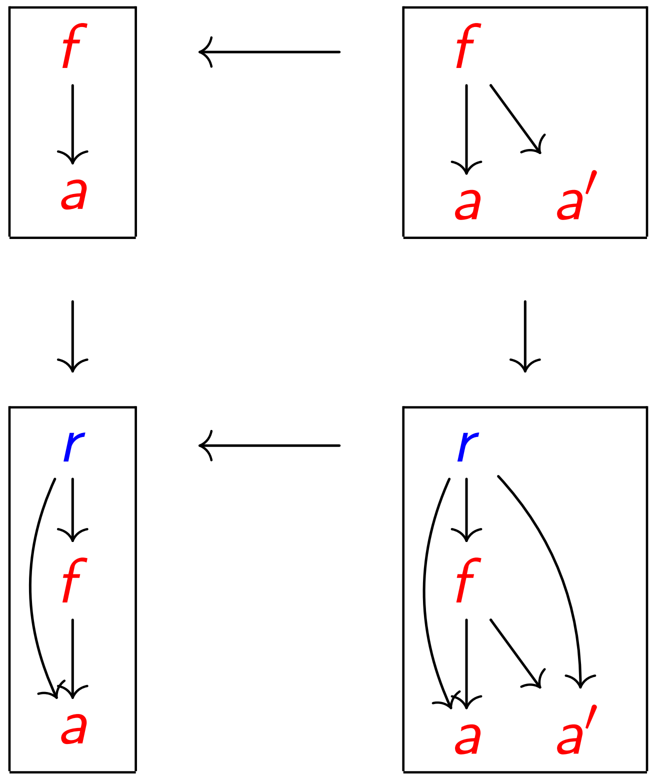
is a pushout (PO, "generalized union")

$$\begin{tikzcd} \boxed{L} \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathrm{PO}"] & \boxed{R} \arrow[d] \\ \boxed{G} \arrow[r] & \boxed{H} \end{tikzcd}$$

Graph transformation: DELETE



Graph transformation: COPY



Graph transformation: DELETE and COPY

- A rule $L \rightsquigarrow R$ is a graph homomorphism $L \leftarrow R$ from R to L
- A step

$$\begin{tikzcd} \boxed{L} \arrow[r, rightsquigarrow] \arrow[d] & \boxed{R} \arrow[d] \\ \boxed{G} \arrow[r, rightsquigarrow] & \boxed{H} \end{tikzcd}$$

is a *?? kind of converse of pushout ??*

$$\begin{tikzcd} \boxed{L} \arrow[d] \arrow[dr, phantom, "??"] & \boxed{R} \arrow[l] \arrow[d] \\ \boxed{G} & \boxed{H} \arrow[l] \end{tikzcd}$$

Algebraic graph transformation

Algebraic graph rewriting is based on:

- some kind of "converse of pushout" (DELETE and COPY)
- followed by a pushout (ADD and MERGE)

$$\begin{tikzcd} L \arrow[d] \arrow[dr, phantom, "??"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, dashed] \arrow[dr, phantom, "PO"] & R \arrow[d, dotted] \\ G & D \arrow[l, dashed, "l_1"] \arrow[r, dotted, "r_1"'] & H \end{tikzcd}$$

- Double-pushout: DPO: when ??=POC
  - Variant: Single-pushout: SPO
- Sesqui-pushout: SqPO: when ??=FPBC
  - Generalization of DPO and SPO

By: H. Ehrig, U. Montanari, H.J. Kreowski, M. Löwe, A. Corradini, B. König, F. Orejas, L. Ribeiro, T. Heindel, F. Hermann, U. Golas, ...

FASE 2014

D. Duval, R. Echahed, F. Prost.

- TERMGRAPH 2006
- RTA 2007
- RTA 2009
- GT-VMT 2011
- ICGT 2012

D. Duval, R. Echahed, F. Prost, L. Ribeiro.

- FASE 2014

  Fundamental Approaches to Software Engineering

  Transformation of Attributed Structures with Cloning

Outline

Term rewriting

Graph transformation

Attributed graph transformation: FASE 2014

Transformations of Attributed Structures with Cloning

Dominique Duval, Rachid Echahed, Frederic Prost, Leila Ribeiro

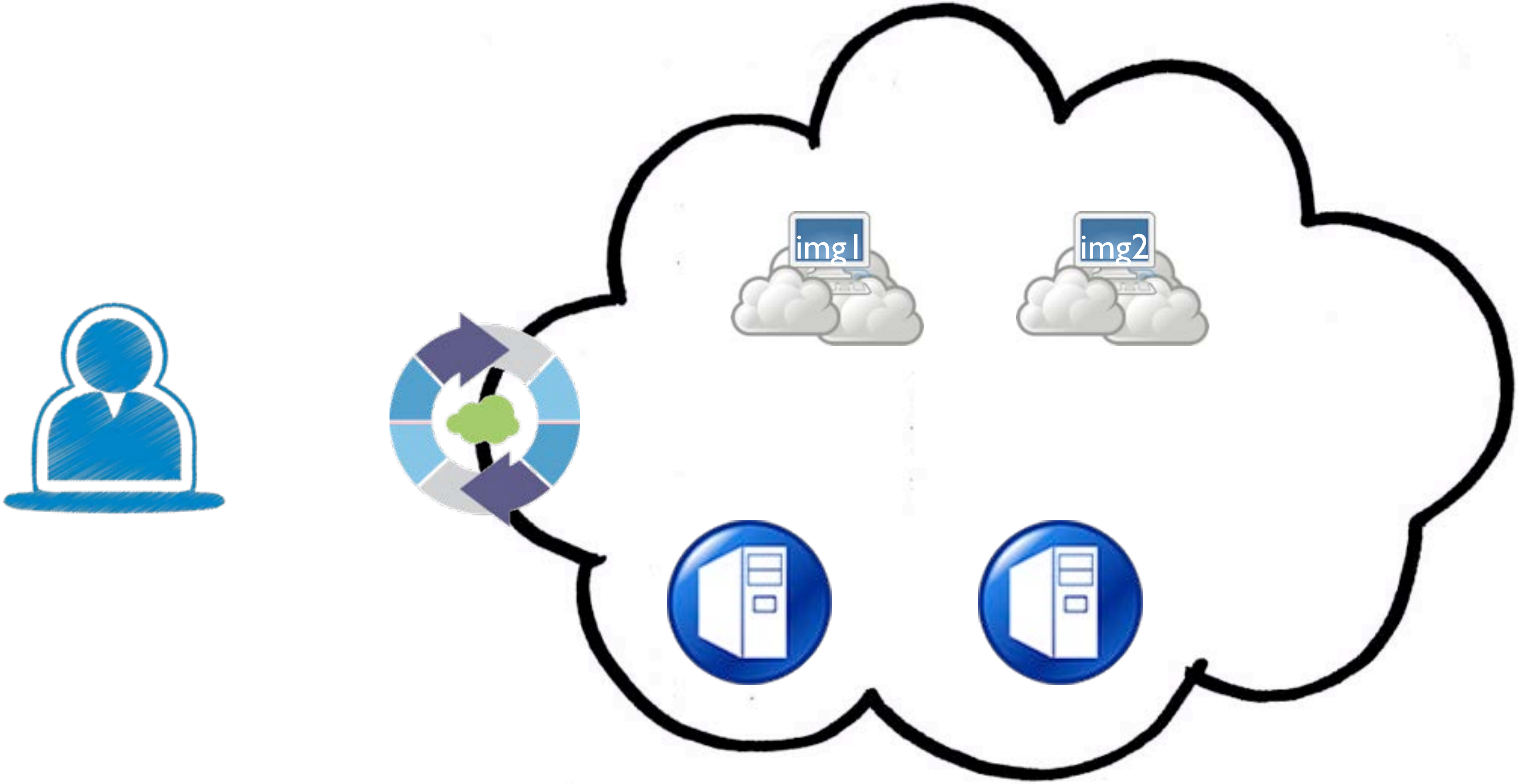
Outline

- Motivation
- Example : Cloud administration
- Attributed Structures
- Sesqui-PO Rewriting of Attributed Structures
- Conclusion and Future work

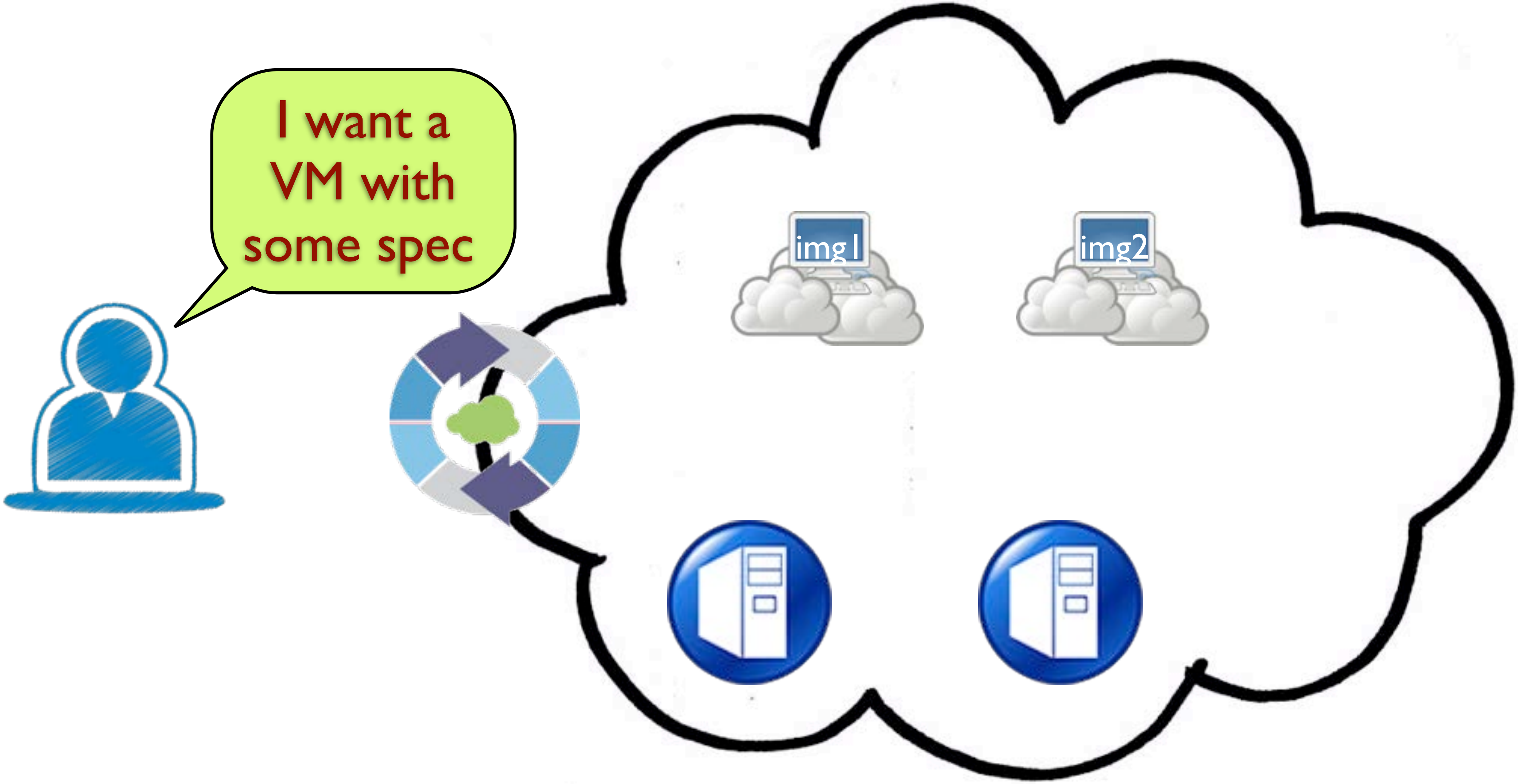
Motivation

- Simple but generic attribute notion
- Cloning possibility

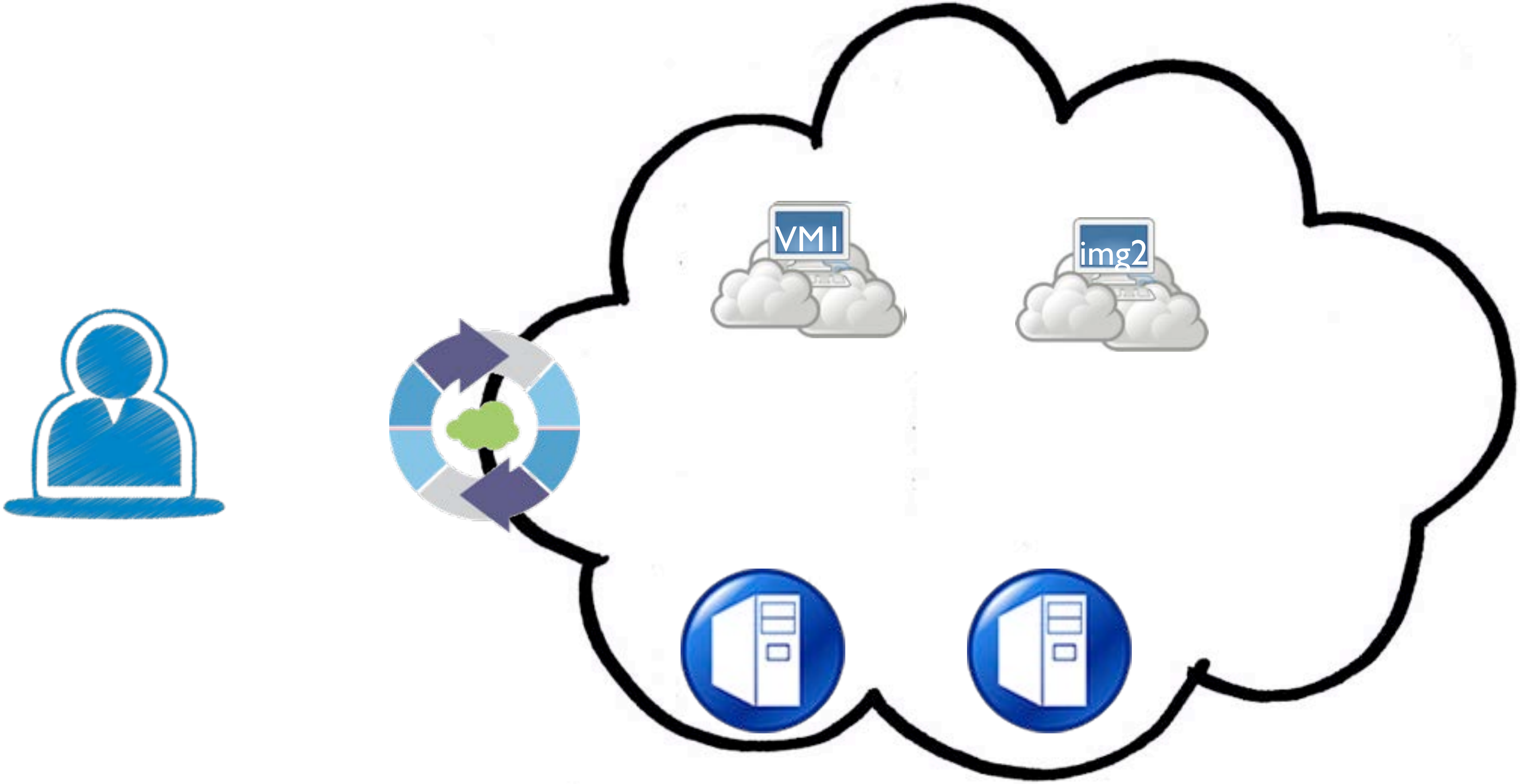
Example: Cloud Adm



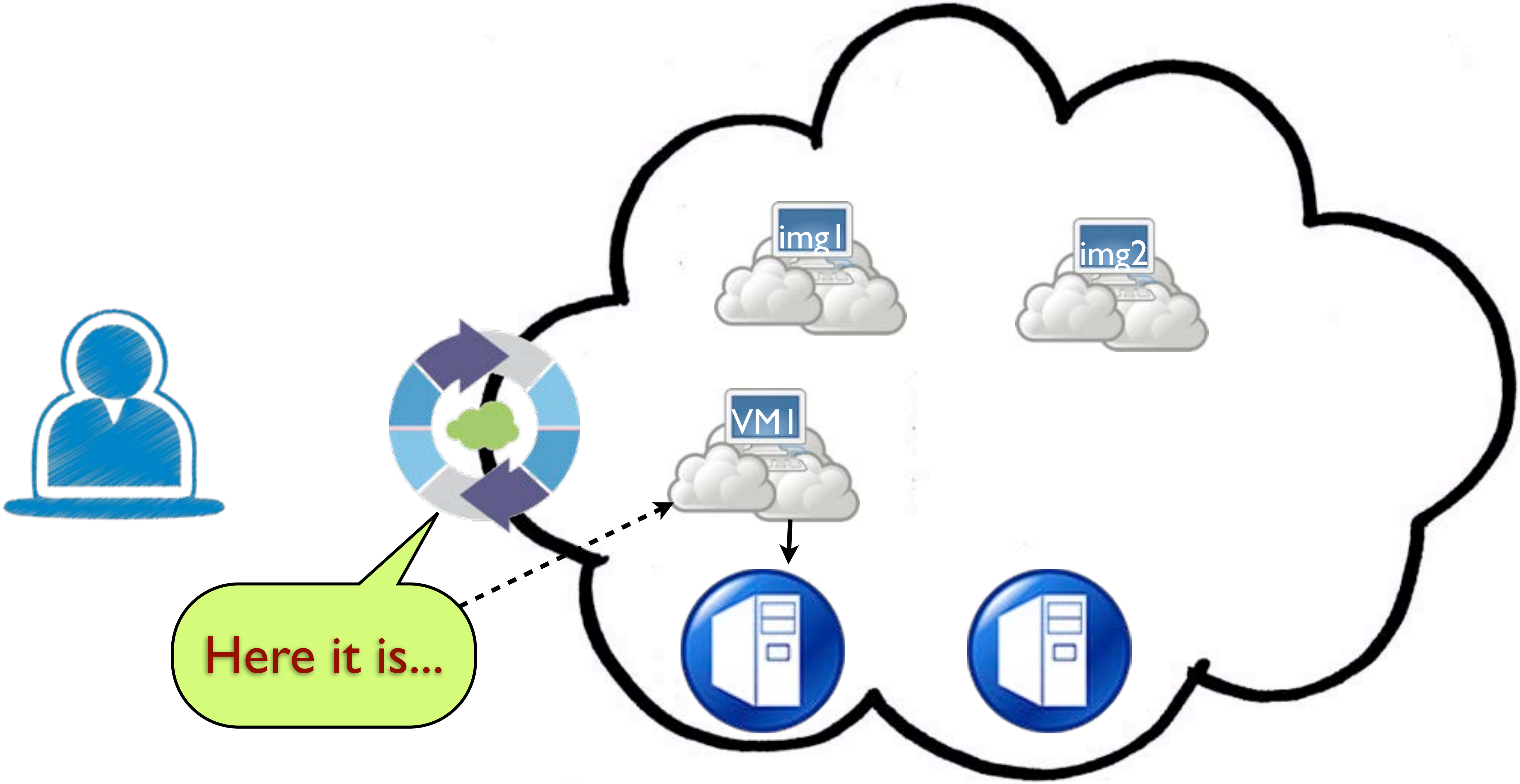
Example: Cloud Adm



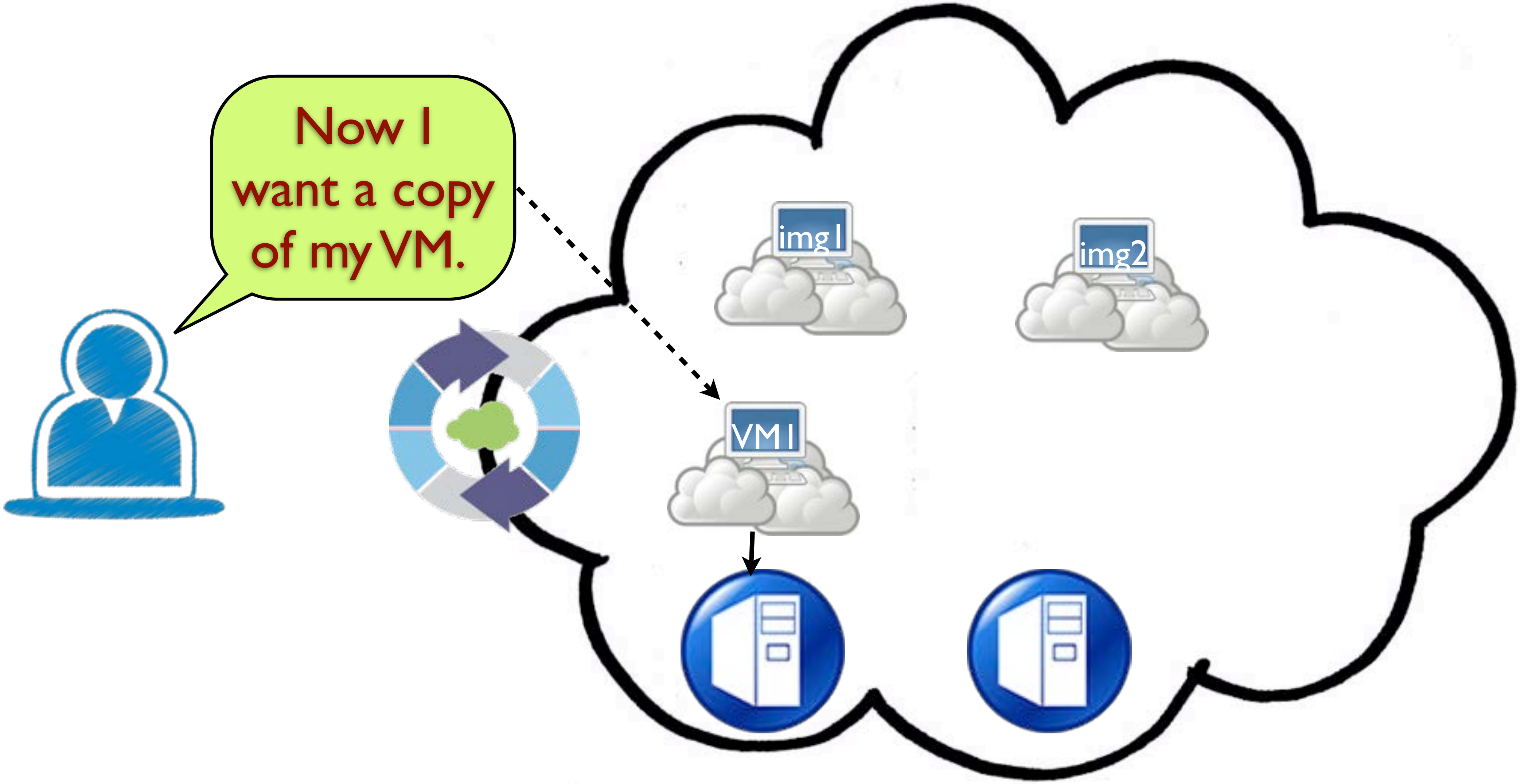
Create VM



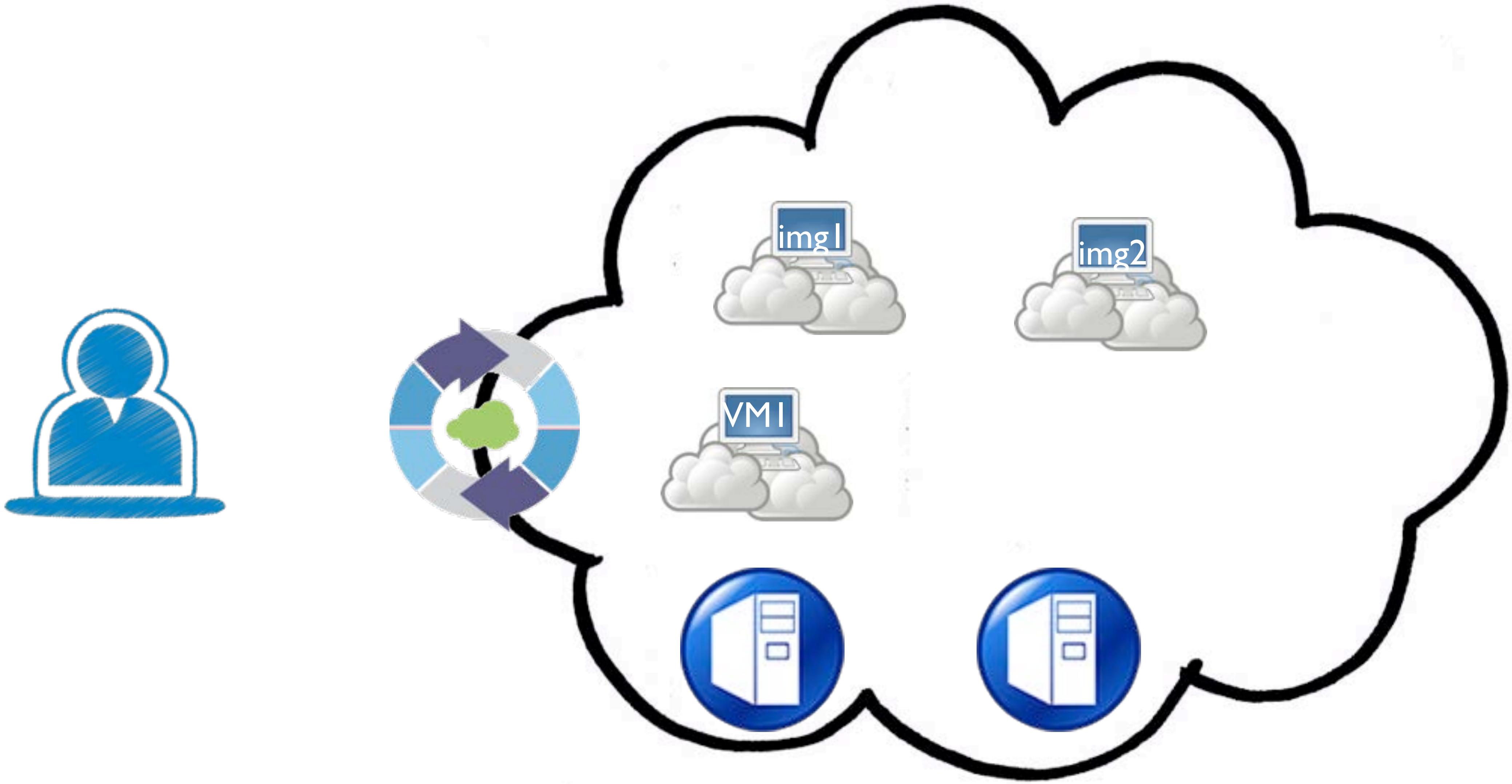
Create VM



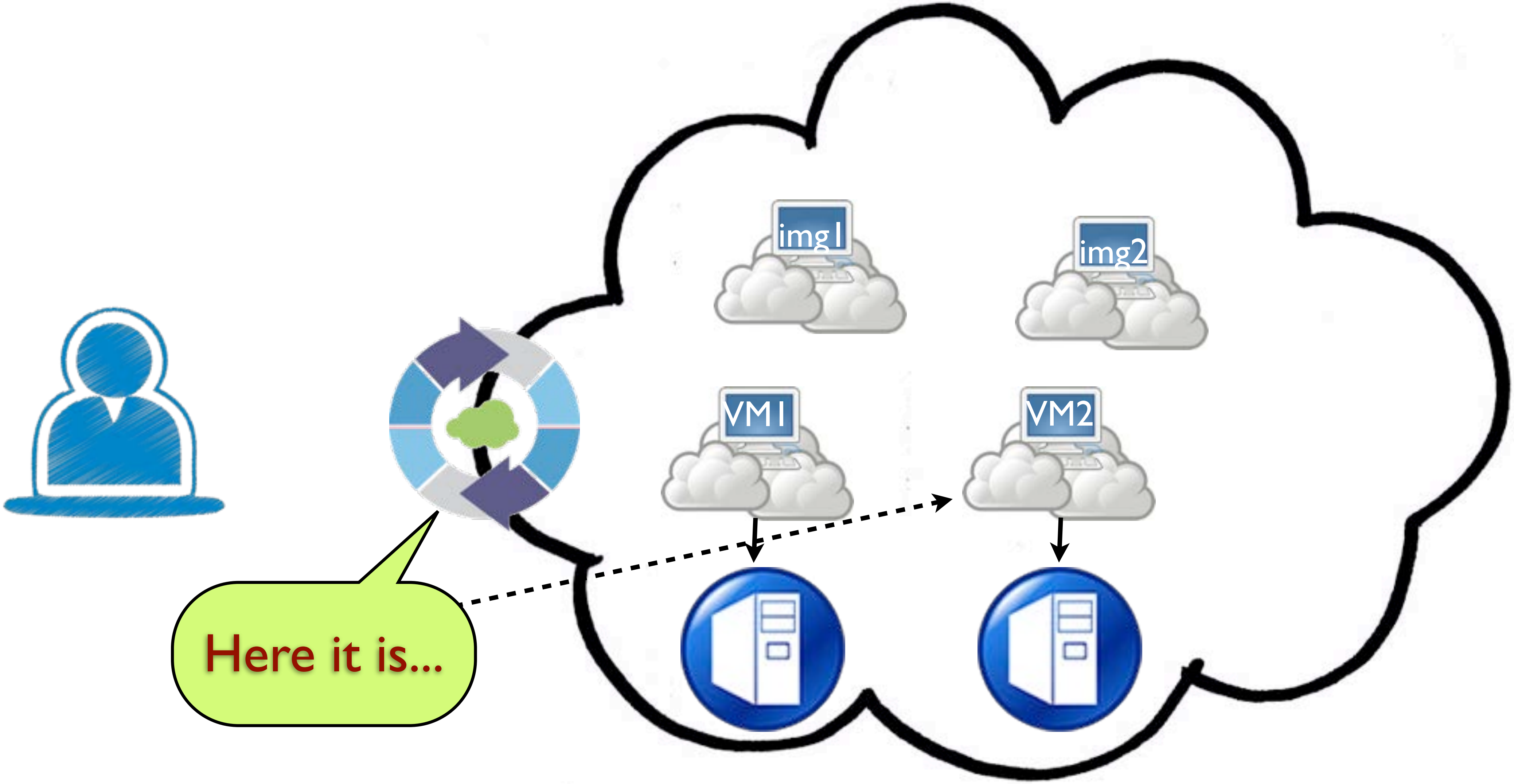
New request...



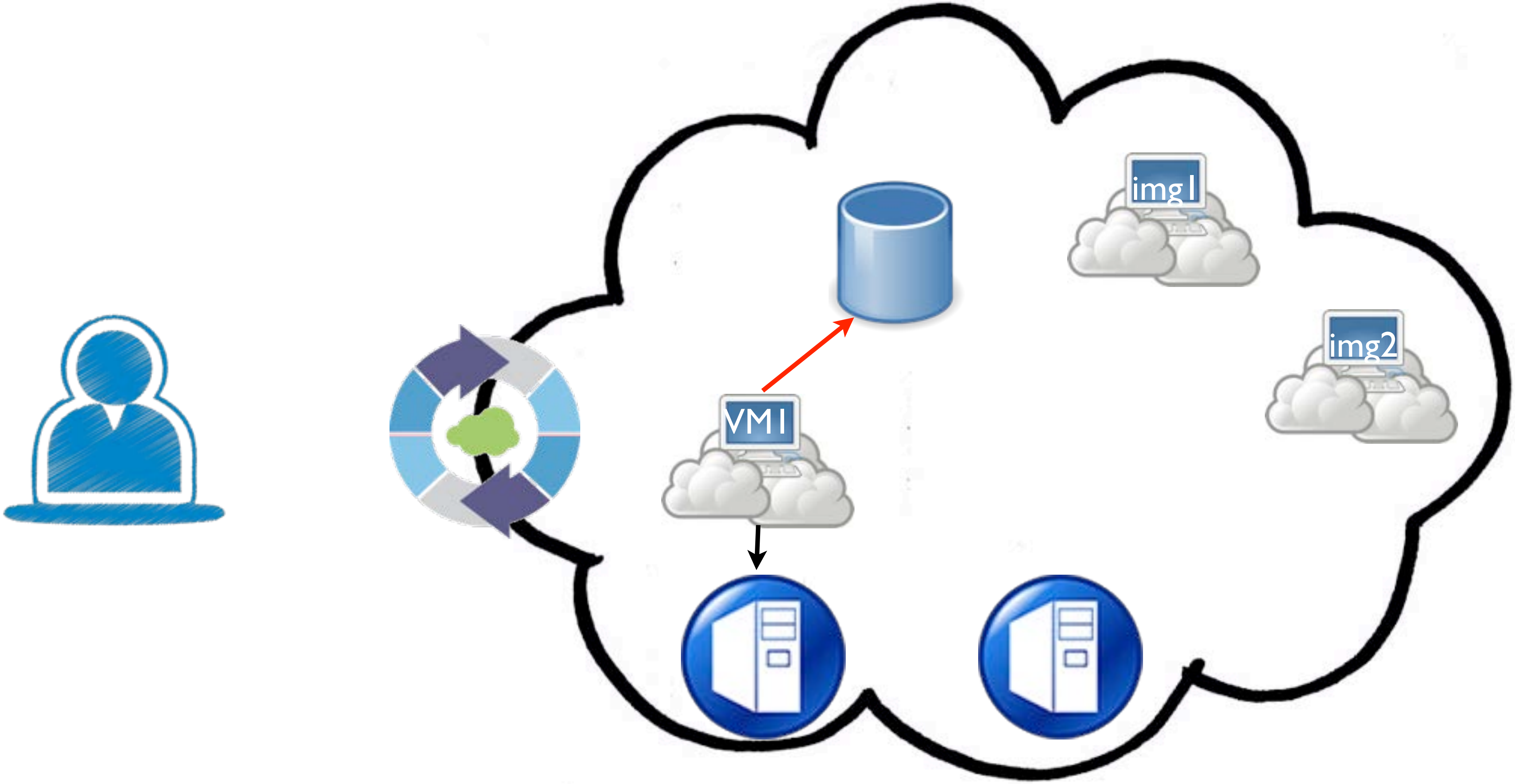
Replicate VM



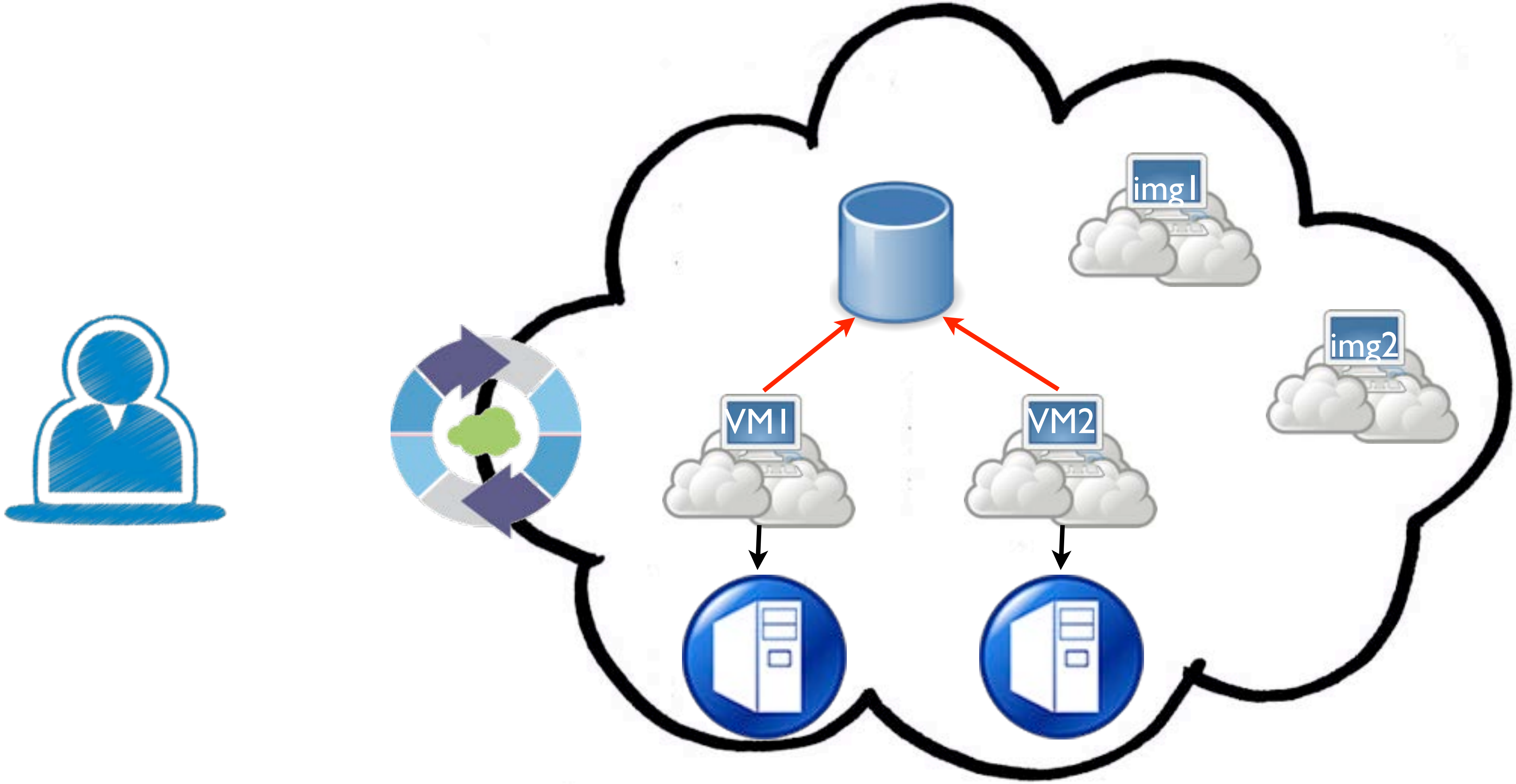
Replicate VM



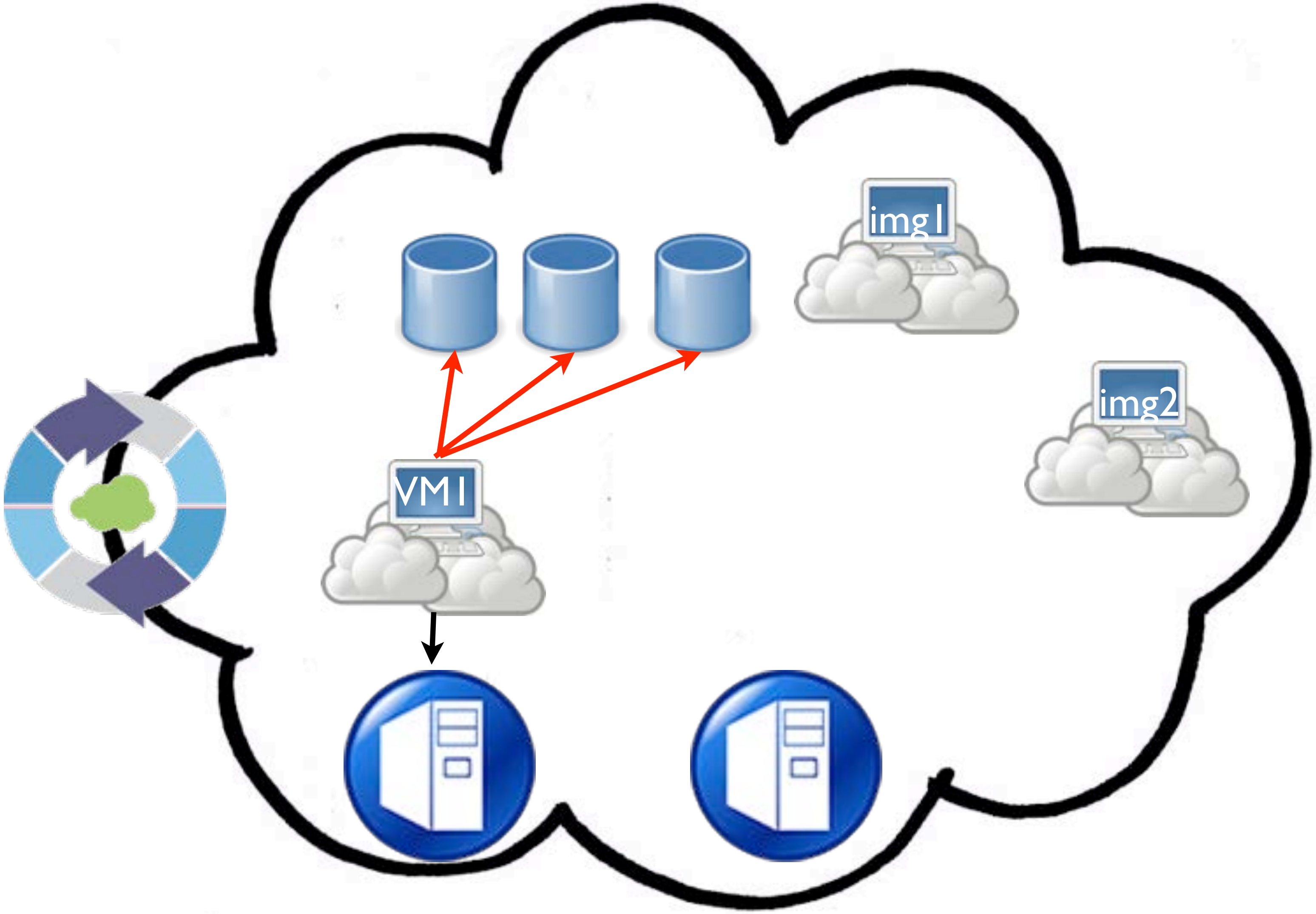
Replicate VM



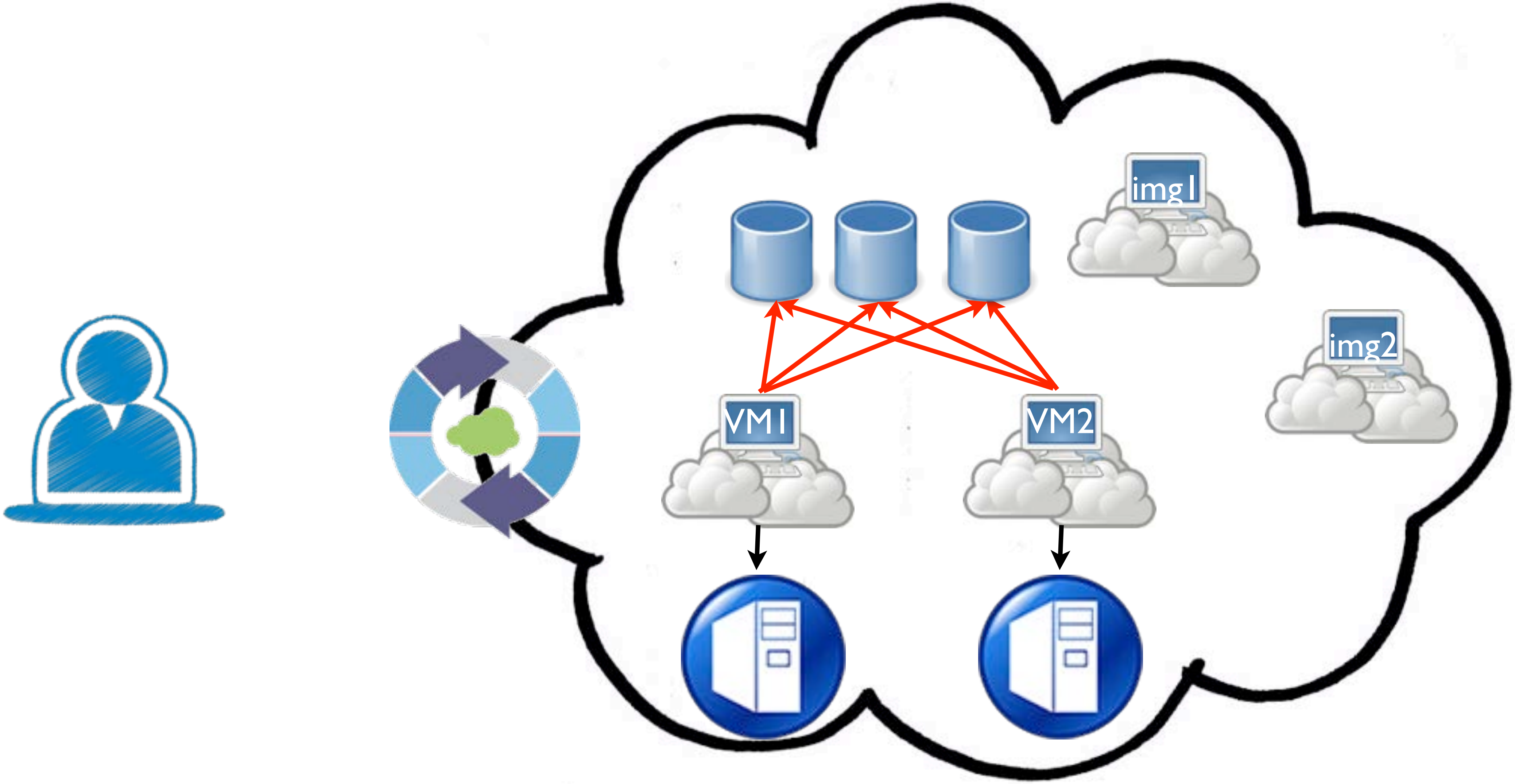
Replicate VM



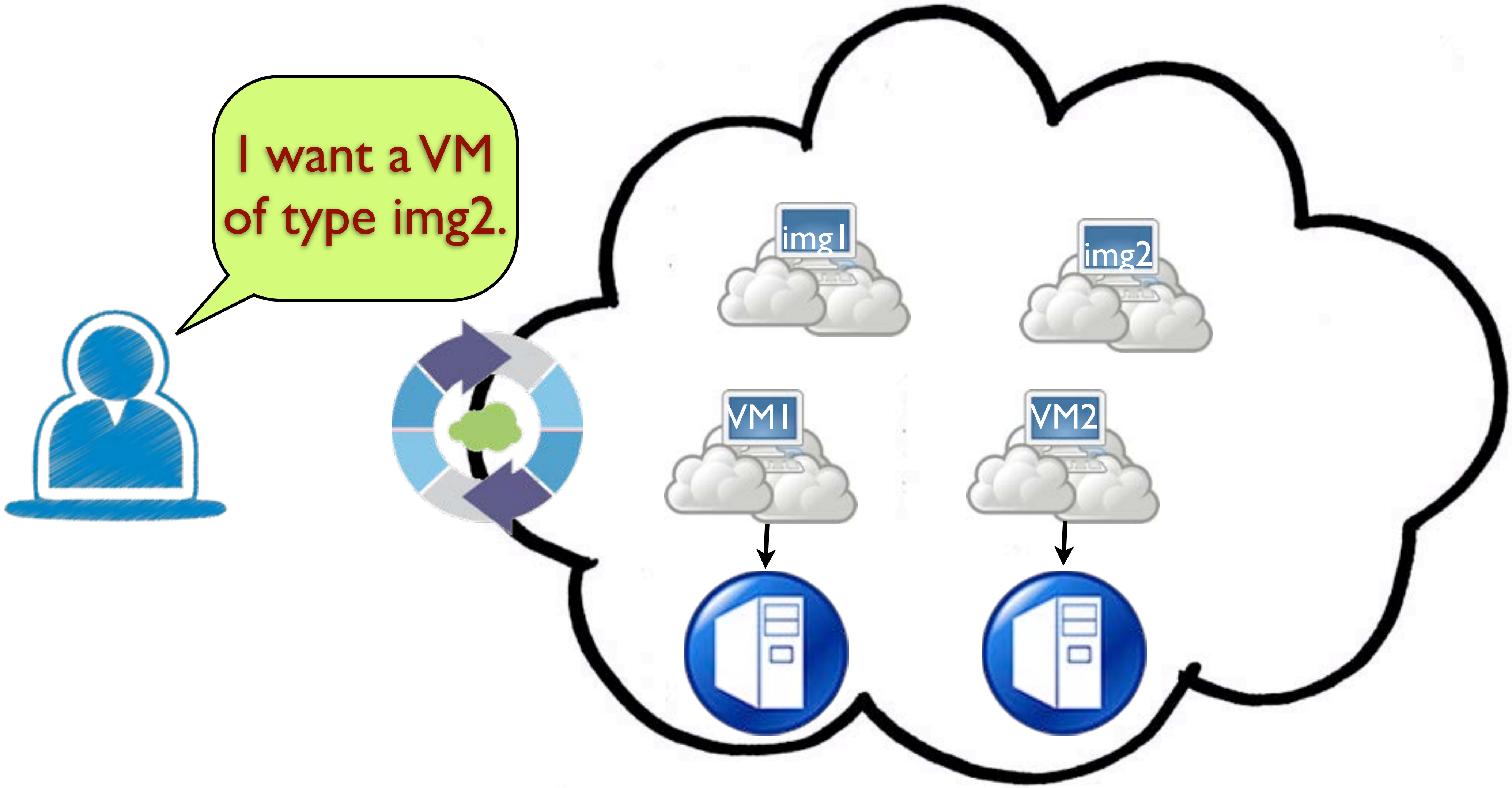
Replicate VM



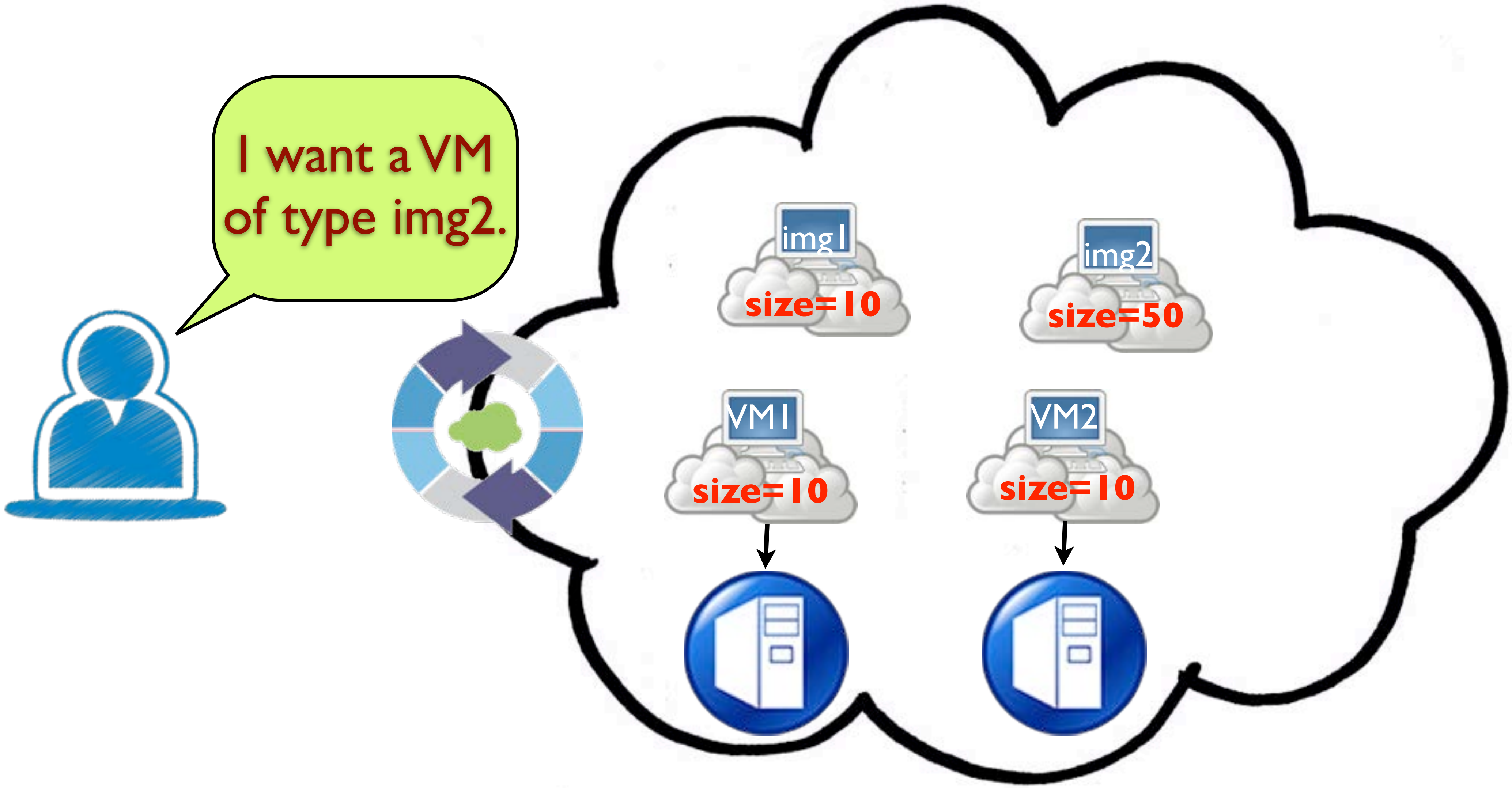
Replicate VM



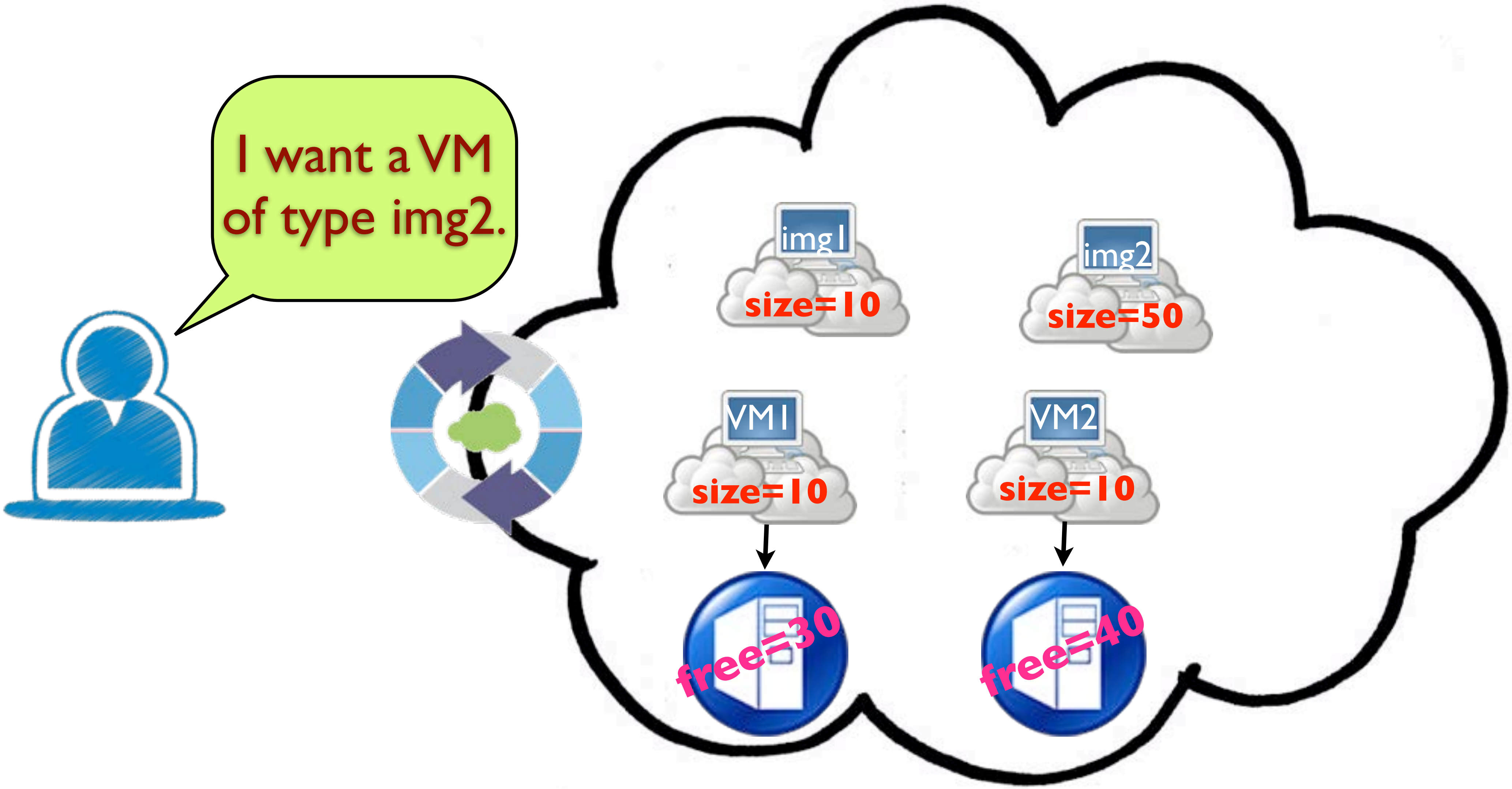
Cloud needs more Mch



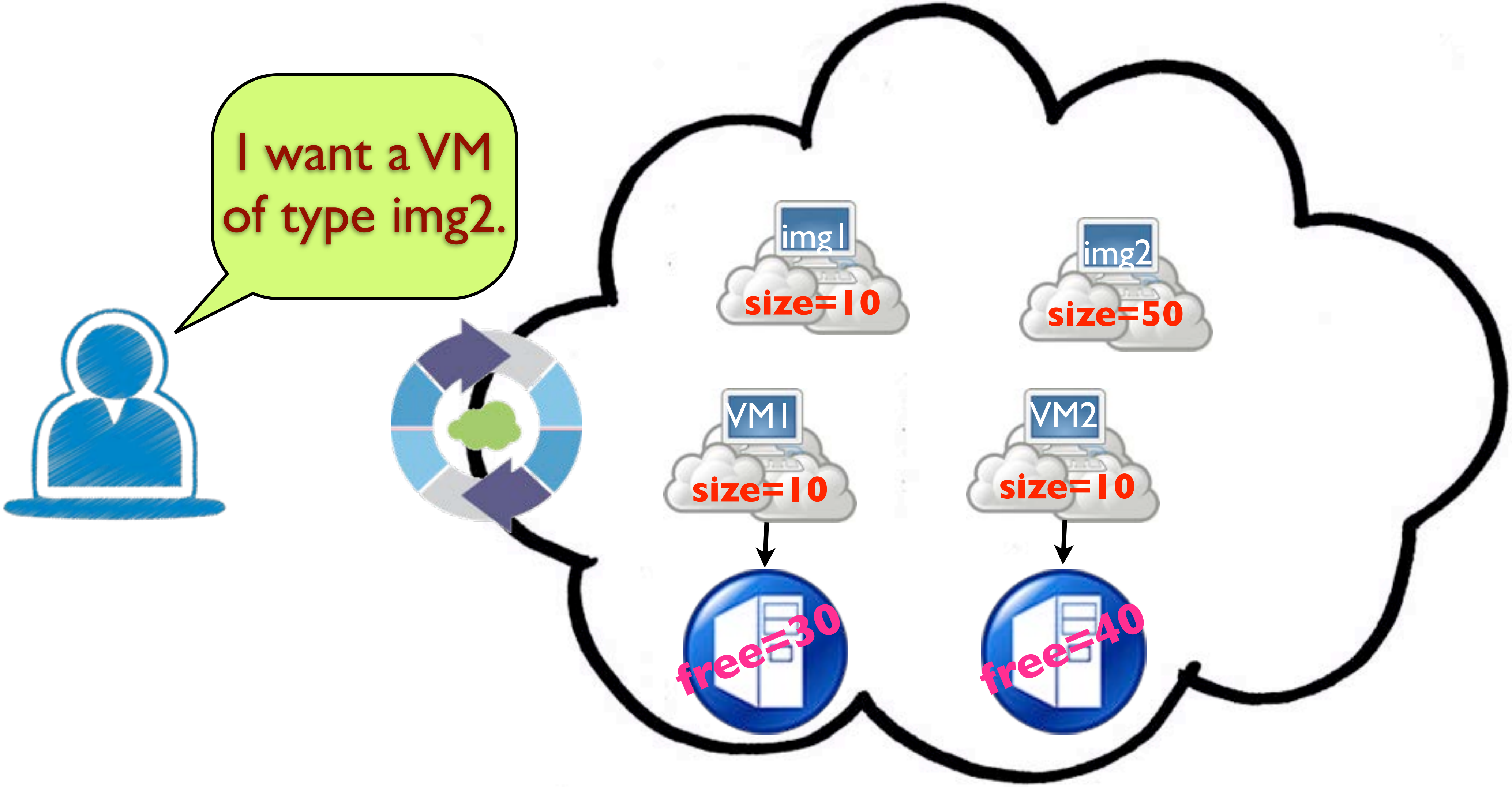
Cloud needs more Mch



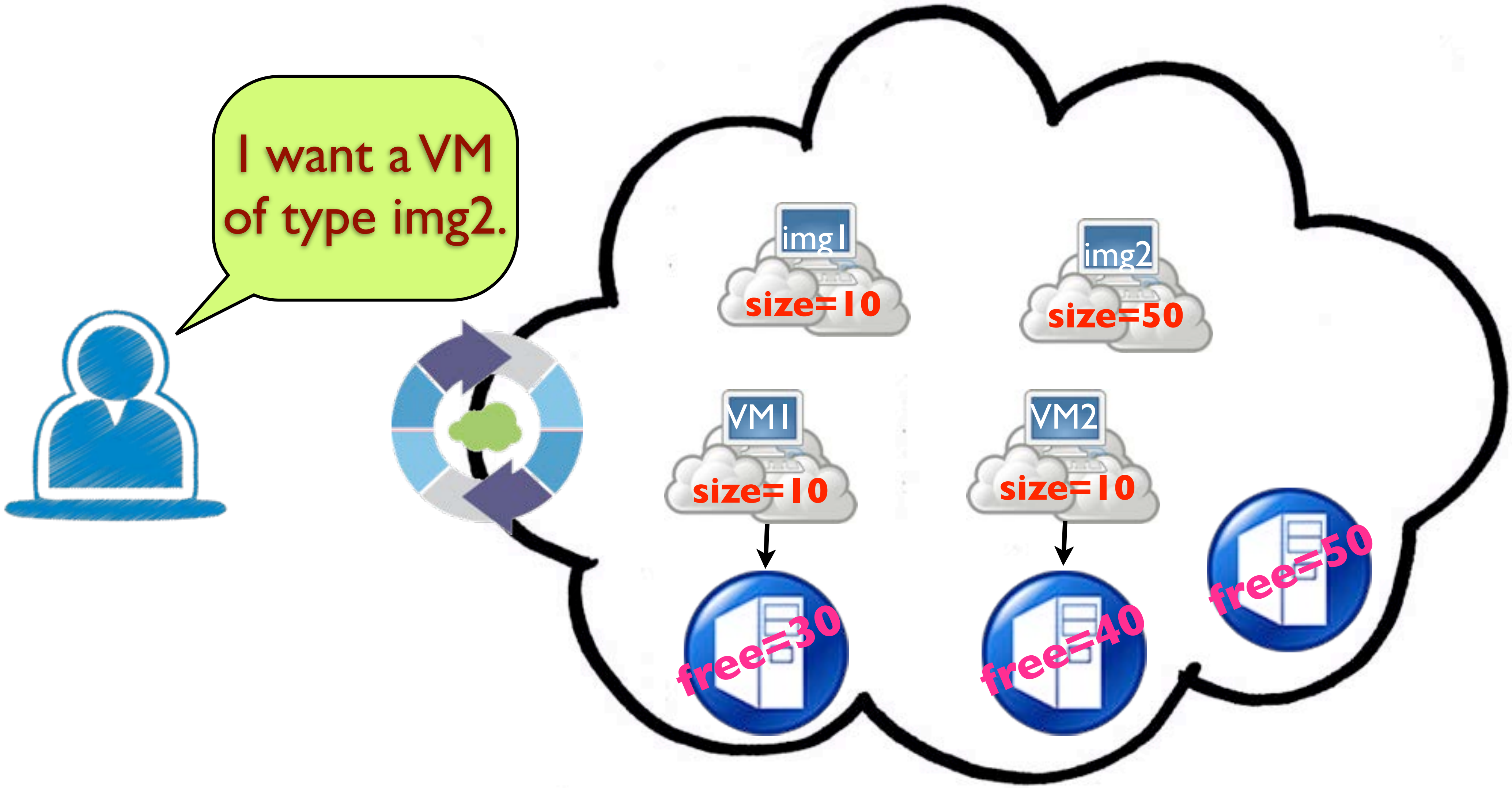
Cloud needs more Mch



Turn on Computer



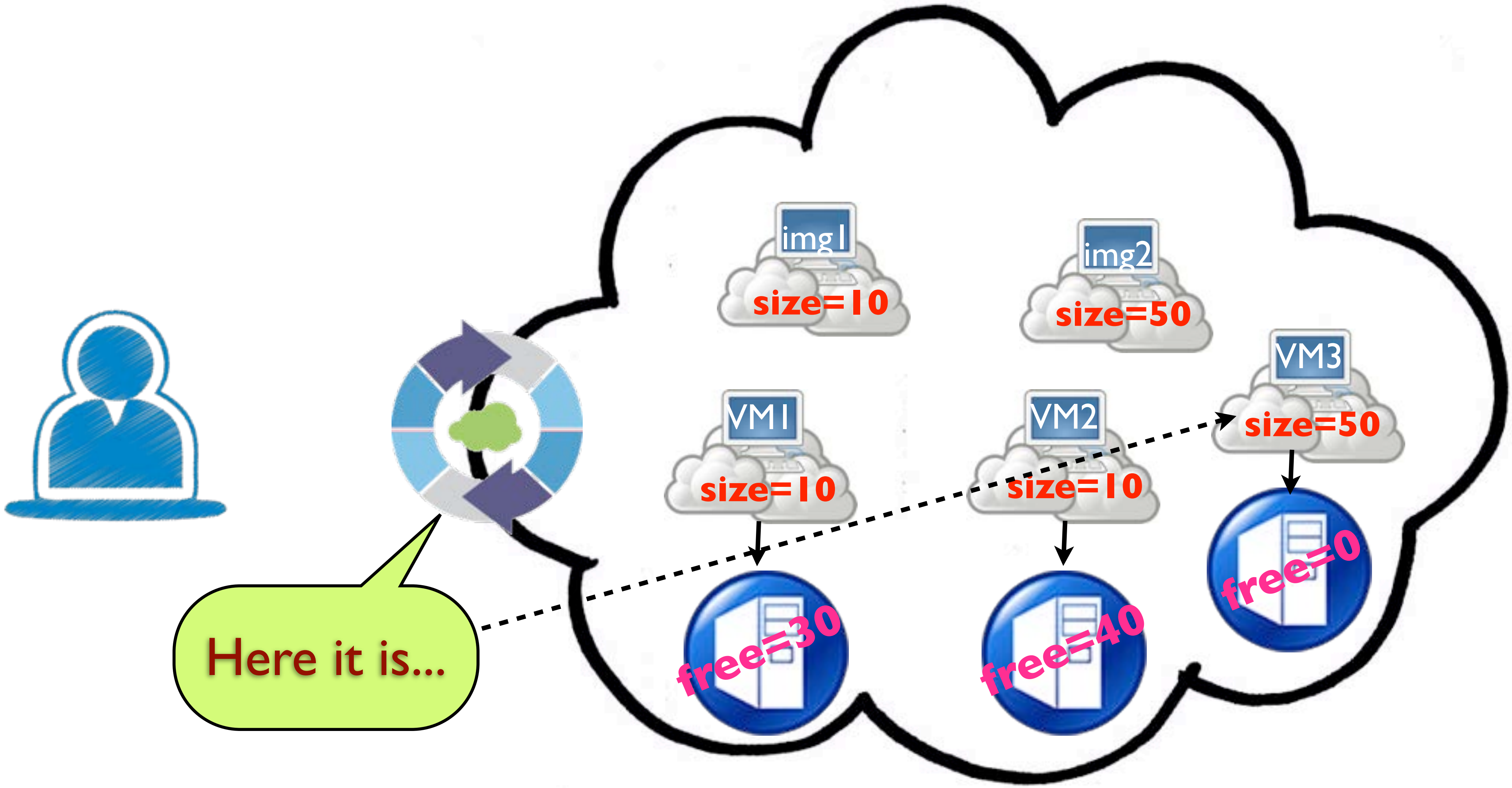
Turn on Computer



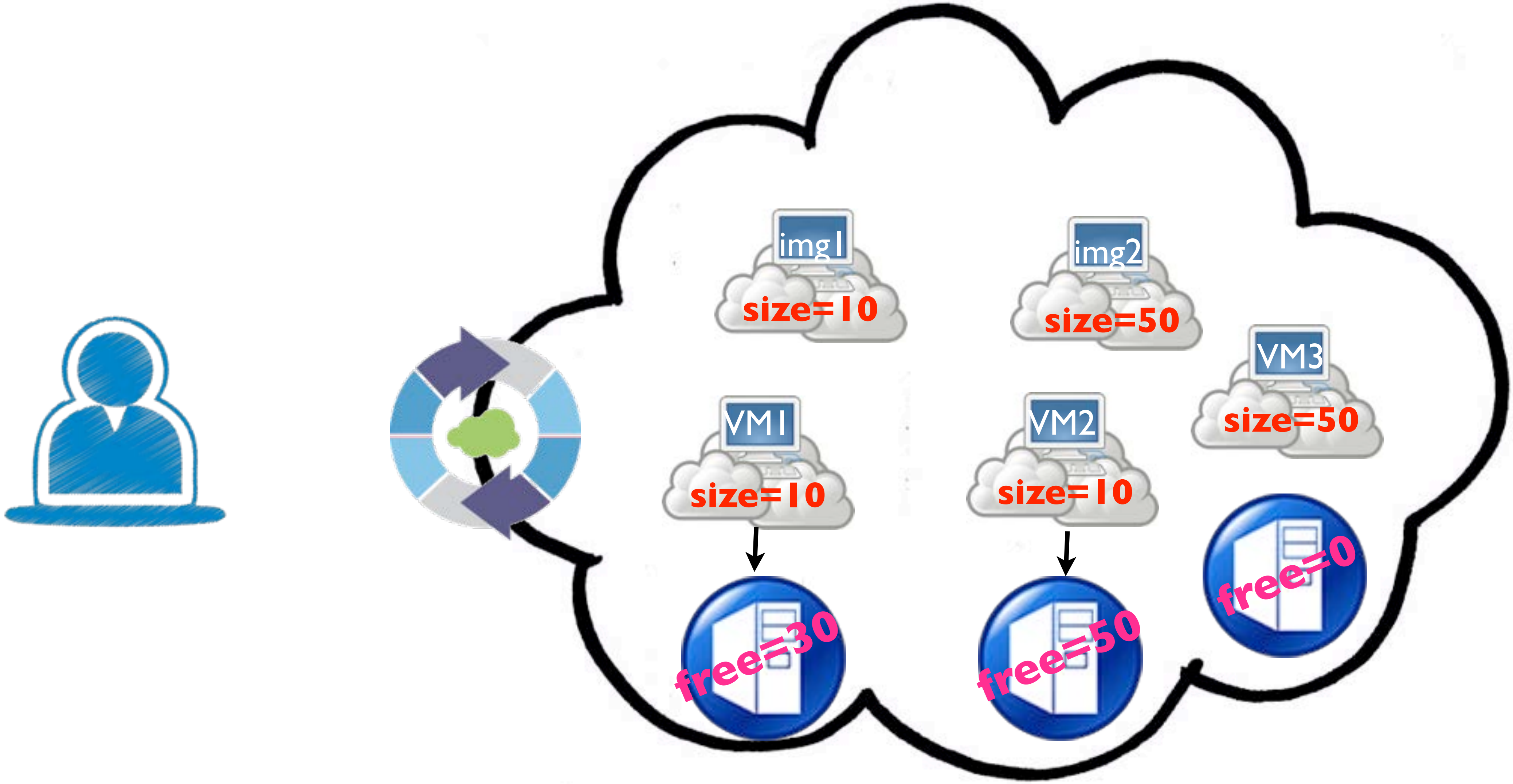
Create VM


img1
size=10
img2
size=50
VM1
size=10
VM2
size=10
free=30
free=40
free=50

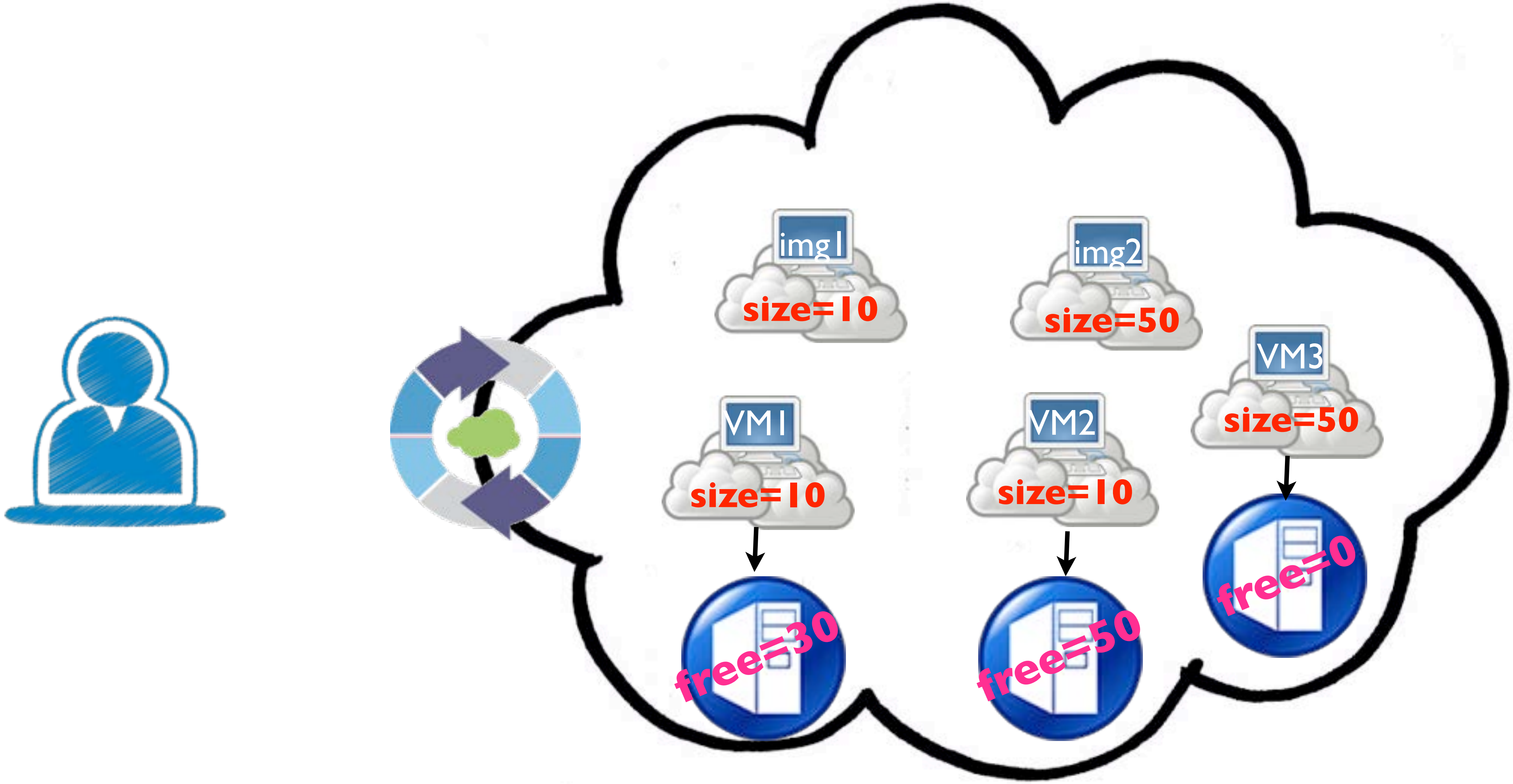
Create VM


img1
size=10
img2
size=50
VM1
size=10
VM2
size=10
VM3
size=50
free=30
free=40
free=0
Here it is...

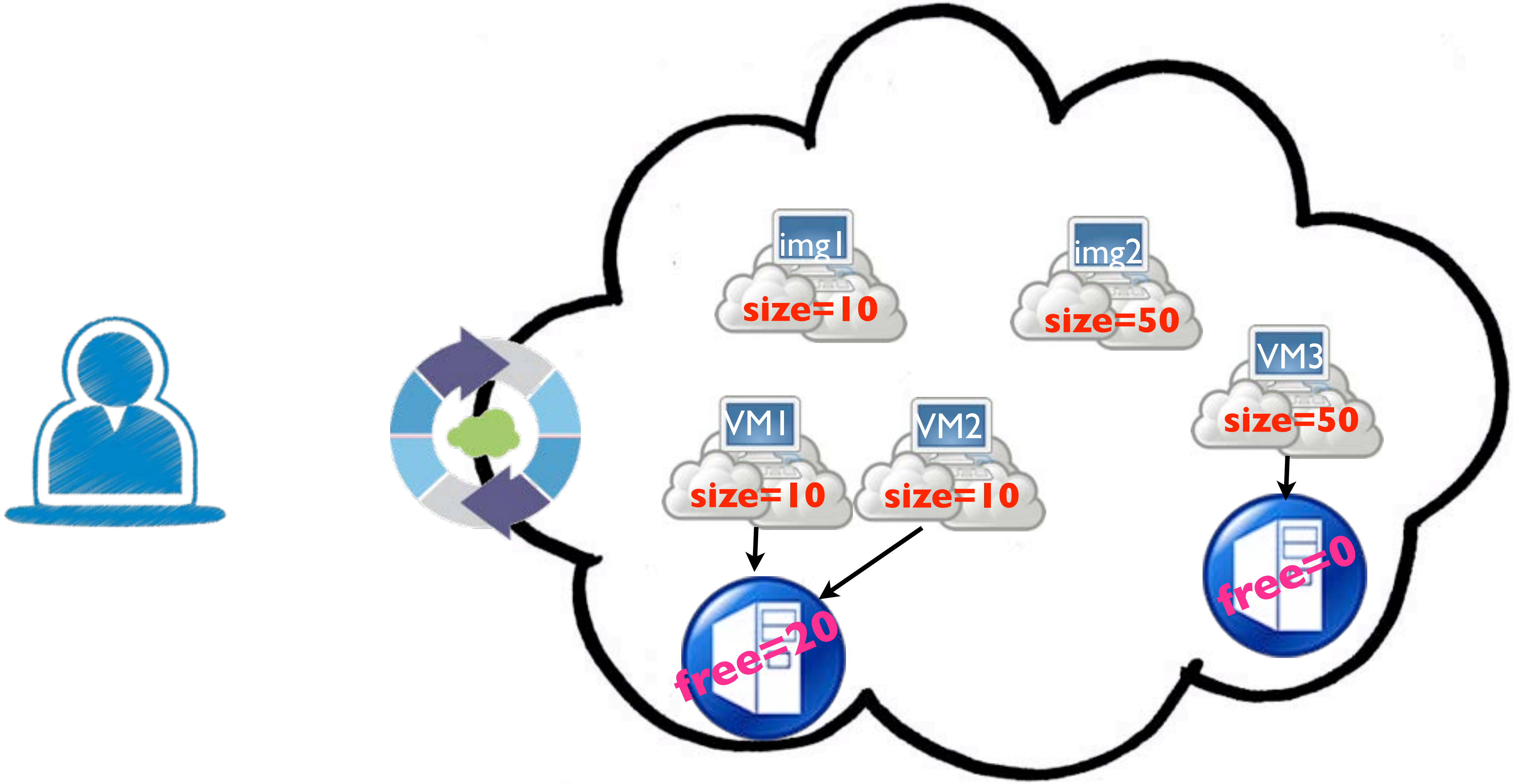
Turn Off Machine



Turn Off Machine



Turn Off Machine



Attributed Structures

Structures

S* : *G* → *Set

Attributes

T* : *A* → *Set

***AttG* = (*S* ↓ *T*)**

$$
\begin{tikzcd}
\widehat{G} \arrow[d, "\widehat{g}"'] \\
\widehat{G'}
\end{tikzcd}
=
\begin{tikzcd}
G \arrow[d, "g"'] \\
G'
\end{tikzcd}
\quad
\begin{tikzcd}
SG \arrow[r, "\alpha"] \arrow[d, "Sg"'] \arrow[dr, phantom, "="] & TA \arrow[d, "Ta"] \\
SG' \arrow[r, "\alpha'"] & TA'
\end{tikzcd}
\quad
\begin{tikzcd}
A \arrow[d, "a"] \\
A'
\end{tikzcd}
$$

Attributed Structures

Structures

$S : G \to \mathbf{Set}$

Attributes

$T : A \to \mathbf{Set}$

Graph

Alg

λ-terms

$$\textit{AttG} = (S \downarrow T)$$

$$\begin{tikzcd} \widehat{G} \arrow[d, "\widehat{g}"'] \\ \widehat{G'} \end{tikzcd} \quad = \quad \begin{tikzcd} G \arrow[d, "g"'] \\ G' \end{tikzcd} \qquad \begin{tikzcd} SG \arrow[r, "\alpha"] \arrow[d, "Sg"'] \arrow[dr, phantom, "="] & TA \arrow[d, "Ta"] \\ SG' \arrow[r, "\alpha'"] & TA' \end{tikzcd} \qquad \begin{tikzcd} A \arrow[d, "a"] \\ A' \end{tikzcd}$$

Attributed Structures

Structures

$S : G \to \mathbf{Set}$

Attributes

$T : A \to \mathbf{Set}$

Graph

Alg

λ-terms

AttG = (S ↓ T)

$$\begin{tikzcd} \widehat{G} \arrow[d, "\widehat{g}"'] \\ \widehat{G'} \end{tikzcd} = \begin{tikzcd} G \arrow[d, "g"'] \\ G' \end{tikzcd} \quad \begin{tikzcd} SG \arrow[r, "\alpha"] \arrow[d, "Sg"'] \arrow[dr, phantom, "="] & TA \arrow[d, "Ta"] \\ SG' \arrow[r, "\alpha'"] & TA' \end{tikzcd} \quad \begin{tikzcd} A \arrow[d, "a"] \\ A' \end{tikzcd}$$

Morphisms preserve attributes

Partially Attributed Structures

Structures

S* : *G* → *Part

Attributes

T* : *A* → *Part

***PAttG* = (*S* ⇣ *T*)**

$$
\begin{tikzcd}
\widehat{G} \arrow[d, "\widehat{g}"'] \\
\widehat{G'}
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
G \arrow[d, "g"'] & S_pG \arrow[r, "\alpha"] \arrow[d, "S_pg"'] \arrow[dr, phantom, "\geq"] & T_pA \arrow[d, "T_pa"] & A \arrow[d, "a"] \\
G' & S_pG' \arrow[r, "\alpha'"] & T_pA' & A'
\end{tikzcd}
$$

Partially Attributed Structures

Structures

S : G → Part

Attributes

T : A → Part

PAttG = (S ⇣ T)

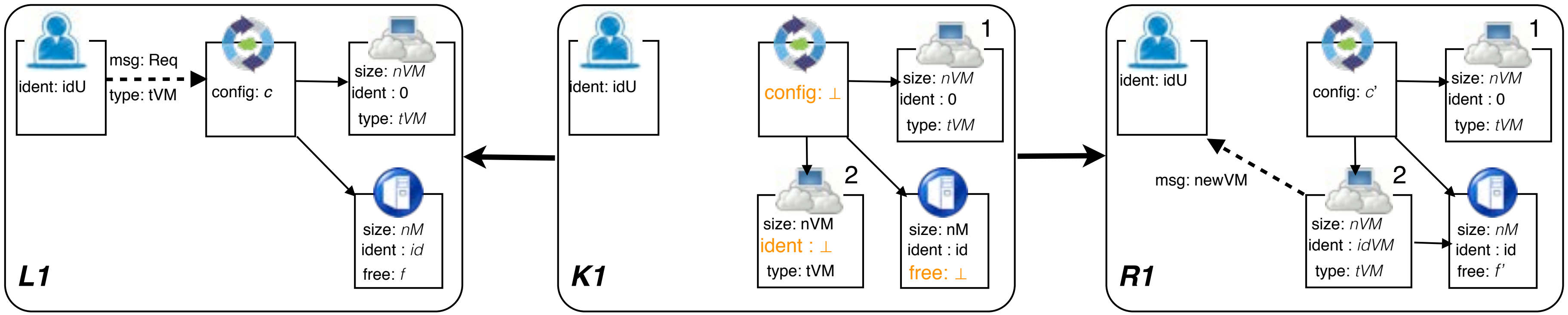
$$
\begin{tikzcd}
\widehat{G} \arrow[d, "\widehat{g}"'] \\
\widehat{G'}
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
G \arrow[d, "g"'] \\
G'
\end{tikzcd}
\quad
\begin{tikzcd}
S_pG \arrow[r, "\alpha"] \arrow[d, "S_pg"'] \arrow[dr, phantom, "\geq"] & T_pA \arrow[d, "T_pa"] \\
S_pG' \arrow[r, "\alpha'"'] & T_pA'
\end{tikzcd}
\quad
\begin{tikzcd}
A \arrow[d, "a"] \\
A'
\end{tikzcd}
$$

Morphisms preserve defined attributes

Graph Transformation Rule

L	←——□K□——→	R
AttrG	**PAttrG**	**AttrG**

rule CreateVM

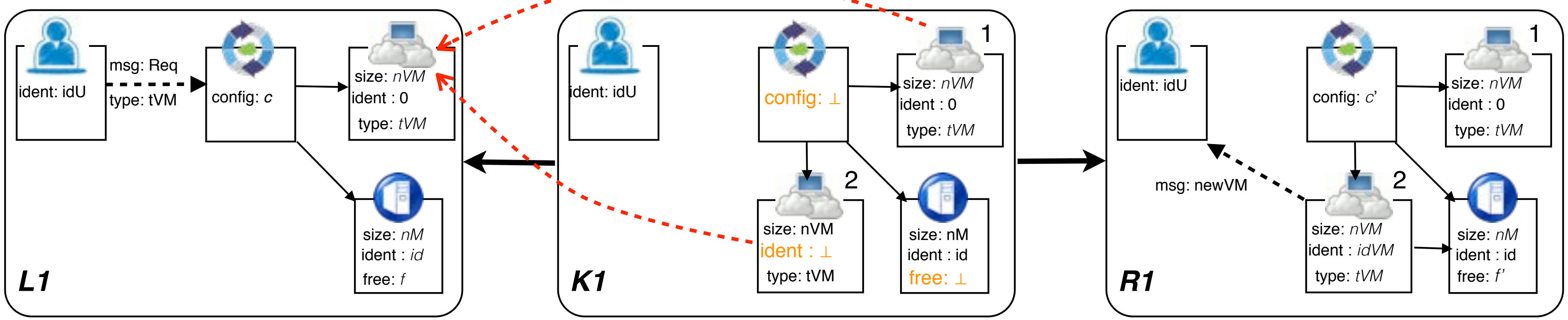


eqns: $idVM = newId(c)$;
$\leq (nVM, f) = true$;
$f' = f - nVM$;
$c' = newVM(c, idU, idVM, nVM, tVM)$

Graph Transformation Rule

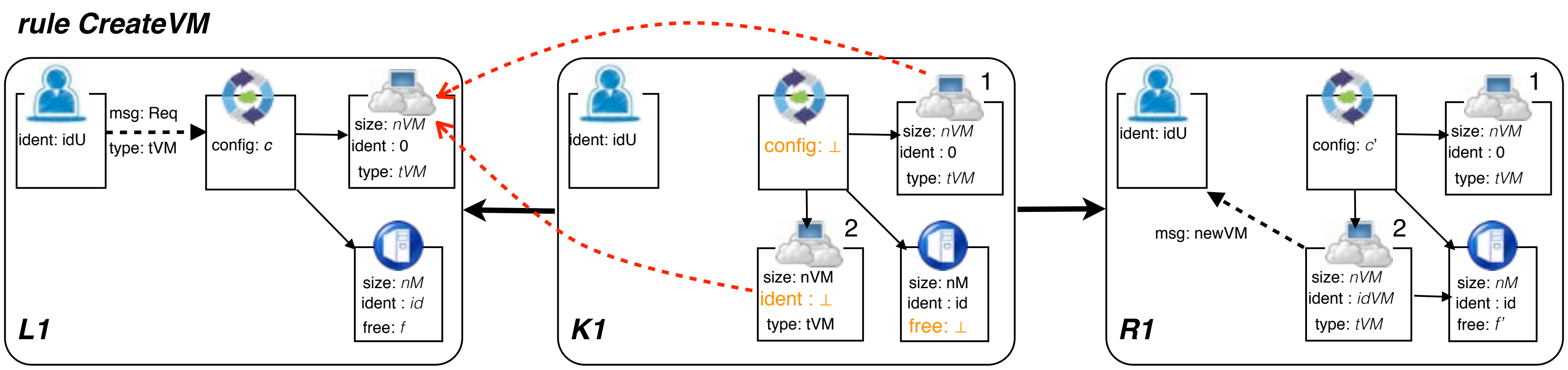
L	← K →	R
AttrG	**PAttrG**	**AttrG**

rule CreateVM

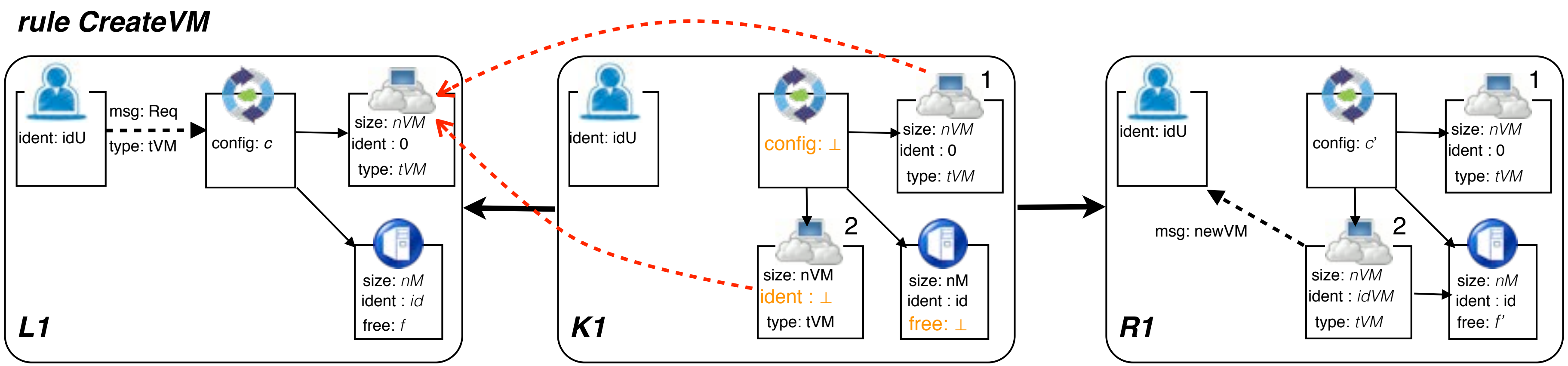


eqns: *idVM = newId(c) ;*
≤ (nVM,f) = true ;
f' = f - nVM ;
c'= newVM(c,idU,idVM,nVM,tVM)

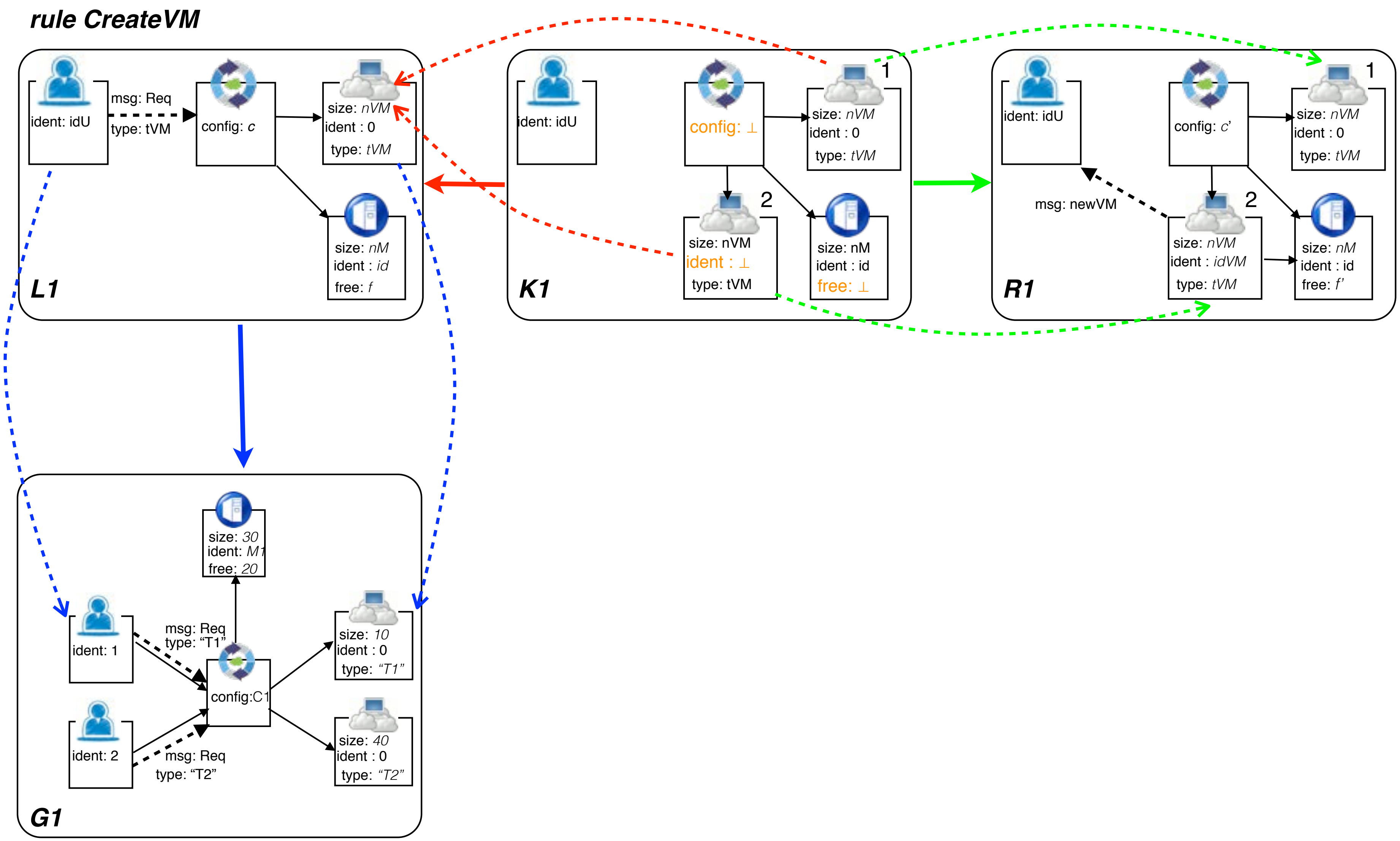
Rule Application



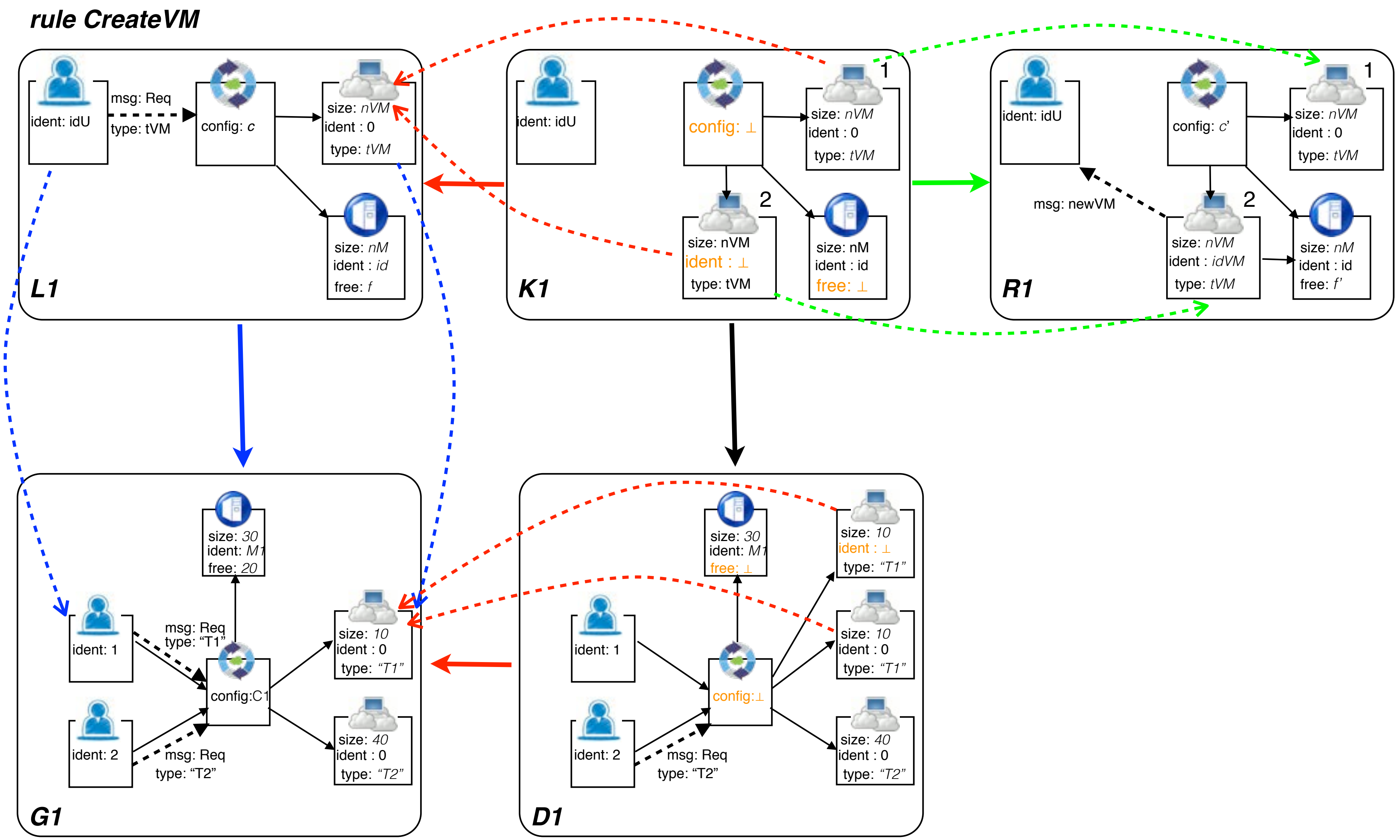
Rule Application



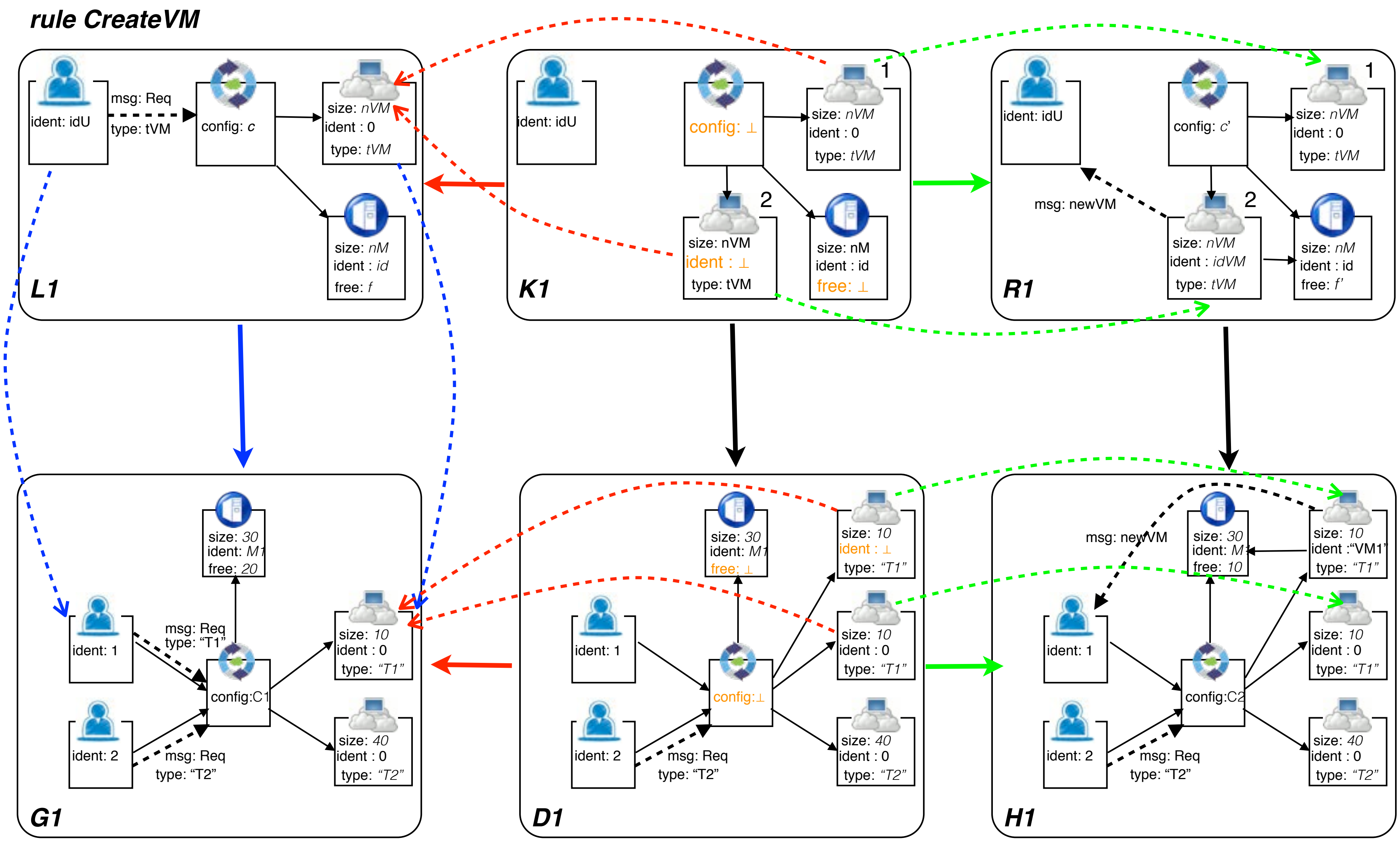
Rule Application



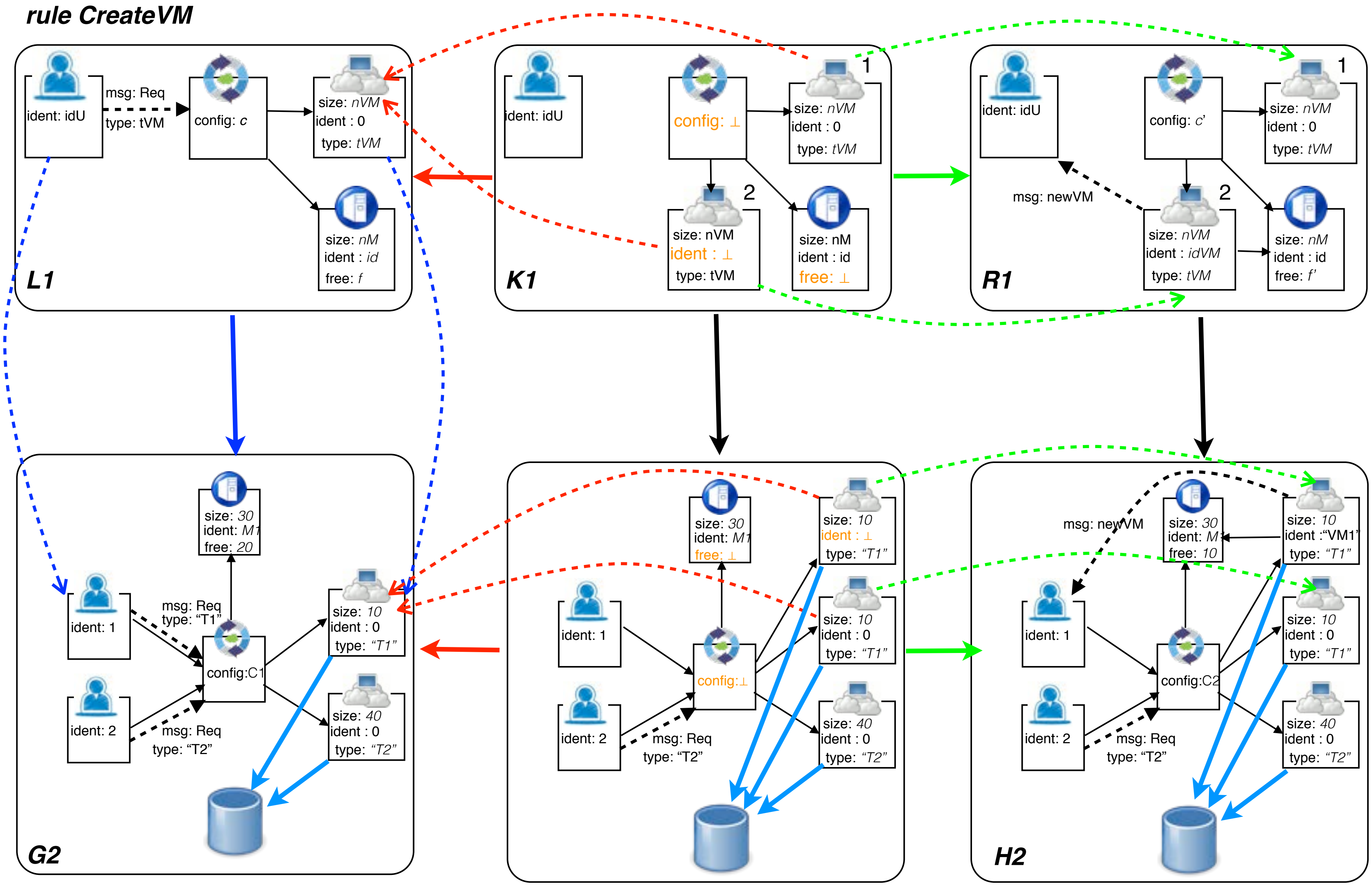
Rule Application



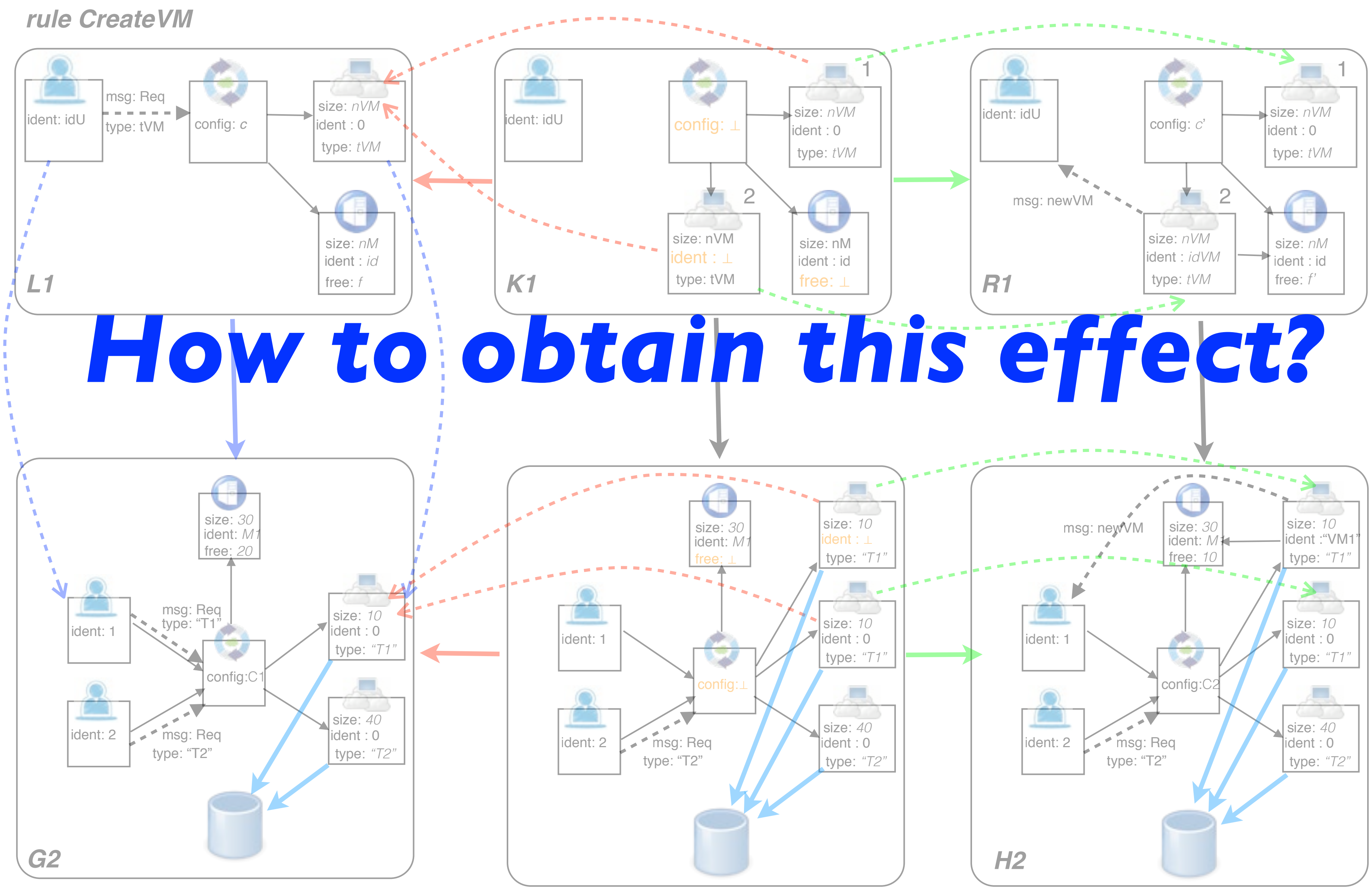
Rule Application



Rule Application



Rule Application



How to obtain this effect?

Sesqui-Pushout Approach

$$\begin{tikzcd} L \arrow[d] \arrow[dr, phantom, "\text{(FPBC)}"] & K \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{(PO)}"] & R \arrow[d] \\ G & D \arrow[l] \arrow[r] & H \end{tikzcd}$$

Final Pullback Complement (FPBC): Deletion and Copy

Pushout (PO): Creation and Merge

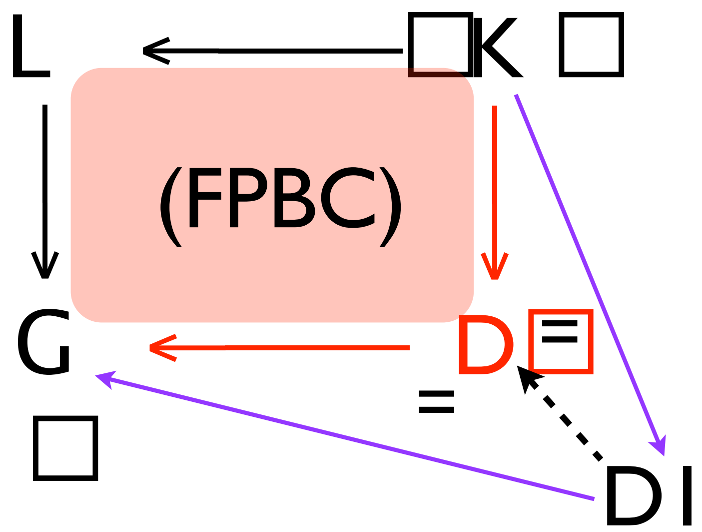
Final Pullback Complement

$$\begin{tikzcd}
L \arrow[d] & K \arrow[l] \\
G &
\end{tikzcd}$$

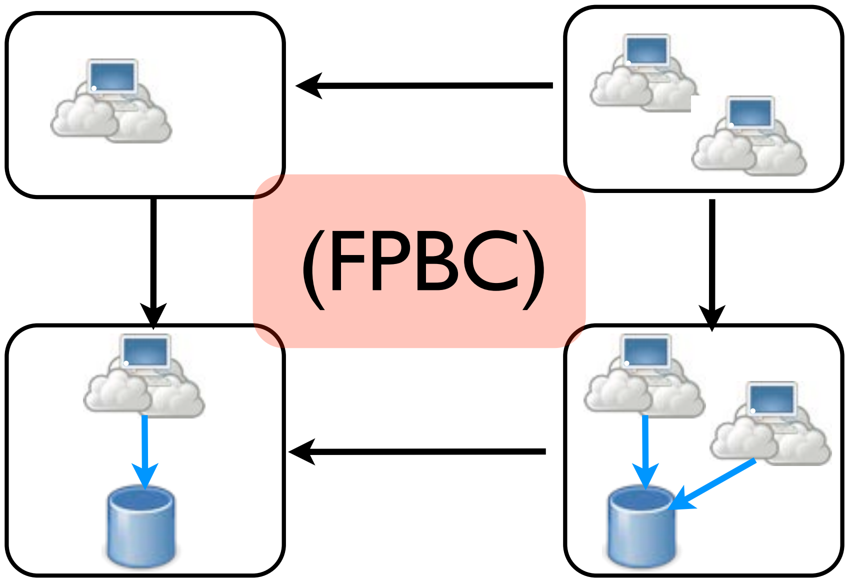
Final Pullback Complement

$$
\begin{tikzcd}
L & \square K \square \arrow[l] \arrow[d] \\
G \square & D \square \arrow[l]
\end{tikzcd}
$$

Final Pullback Complement

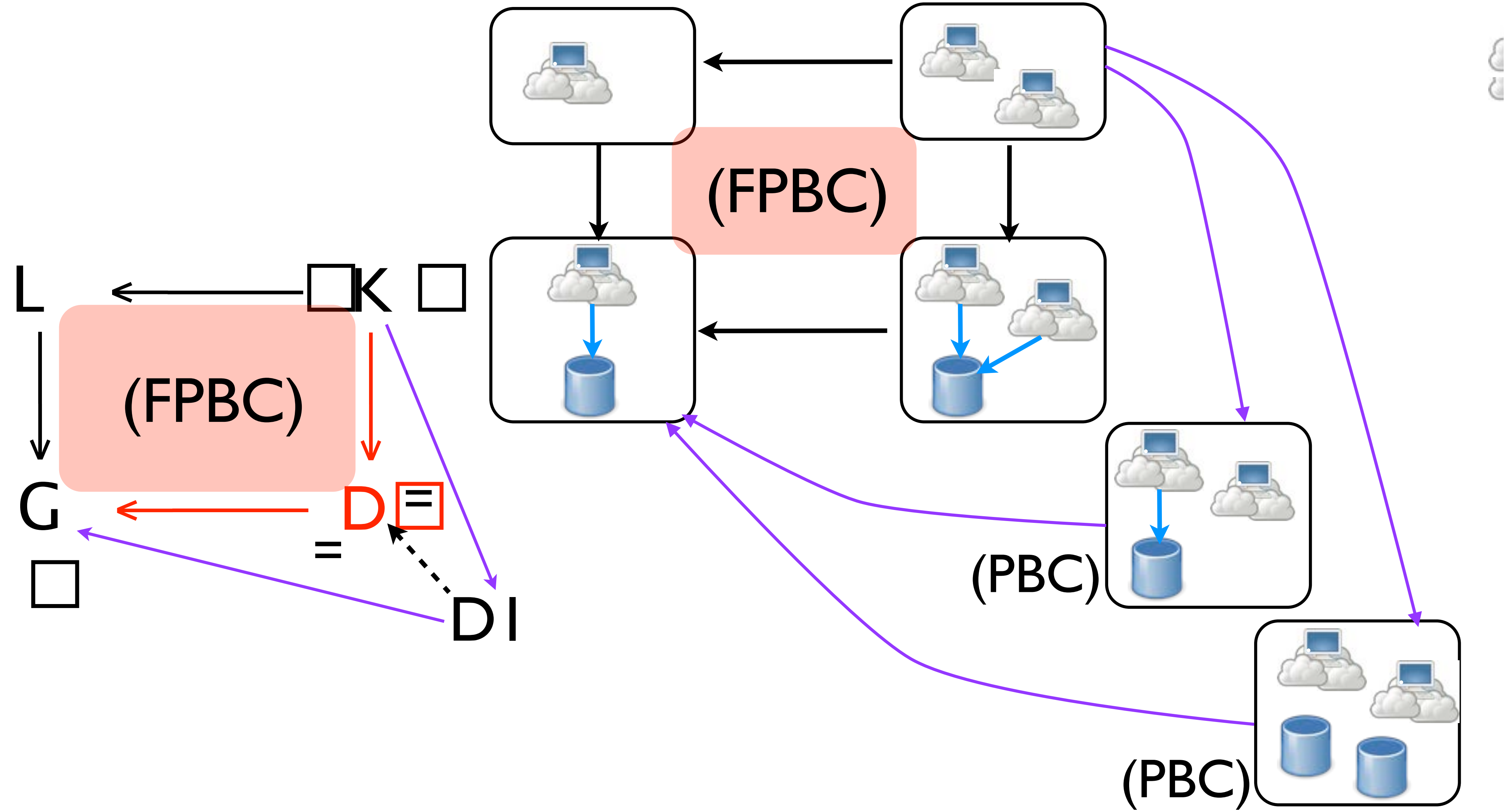


Final Pullback Complement



$$\begin{tikzcd}
L \arrow[d] & K \arrow[l] \arrow[d] \arrow[ddr, bend left=10] \arrow[dl, phantom, "\text{(FPBC)}"] & \\
G & D \arrow[l] \arrow[dl, phantom, "="] & \\
 & & D' \arrow[ull, bend left=10] \arrow[ul, dotted]
\end{tikzcd}$$

Final Pullback Complement



SqPO-Rewriting of Attributed Structures

Structures:

Δ :

$$\begin{tikzcd}
L \arrow[d, "m_L"] \arrow[dr, phantom, "(FPBC)"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "m_K"] \arrow[dr, phantom, "(PO)"] & R \arrow[d, "m_R"] \\
G & D \arrow[l, "l_1"'] \arrow[r, "r_1"] & H
\end{tikzcd}$$

SqPO-Rewriting of Attributed Structures

Structures:

Δ :

$$
\begin{tikzcd}
L \arrow[d, "m_L"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "m_K"] \arrow[dl, phantom, "(FPBC)"] \arrow[dr, phantom, "(PO)"] & R \arrow[d, "m_R"] \\
G & D \arrow[l, "l_1"'] \arrow[r, "r_1"] & H
\end{tikzcd}
$$

Attributed Structures:

$\widehat{\Delta}$:

$$
\begin{tikzcd}
\widehat{L} \arrow[d, "{(m_L,a)}"] & \widehat{K} \arrow[l, "{(l,id_A)}"'] \arrow[r, "{(r,id_A)}"] \arrow[d, "{(m_K,a)}"] \arrow[dl, phantom, "(FPBC)"] \arrow[dr, phantom, "(PO)"] & \widehat{R} \arrow[d, "{(m_R,a}"] \\
\widehat{G} & \widehat{D} \arrow[l, "{(l_1,id_{A_1})}"'] \arrow[r, "{(r_1,id_{A_1})}"] & \widehat{H}
\end{tikzcd}
$$

SqPO-Rewriting of Attributed Structures

Attributed Structures:

$\widehat{\Delta}$:

$$
\begin{tikzcd}
\widehat{L} \arrow[d, "{(m_L,a)}"] \arrow[dr, phantom, "(FPBC)"] & \widehat{K} \arrow[l, "{(l,id_A)}"'] \arrow[r, "{(r,id_A)}"] \arrow[d, "{(m_K,a)}"] \arrow[dr, phantom, "(PO)"] & \widehat{R} \arrow[d, "{(m_R,a)}"] \\
\widehat{G} & \widehat{D} \arrow[l, "{(l_1,id_{A_1})}"'] \arrow[r, "{(r_1,id_{A_1})}"] & \widehat{H}
\end{tikzcd}
$$

SqPO-Rewriting of Attributed Structures

x is context: *keep attribute*

$$l_1(x) : t_1 \longleftarrow\!\!\dashv x : t_1 \longmapsto r_1(x) : t_1$$

Attributed Structures:

$\widehat{\Delta}$:

$$\begin{tikzcd}
\widehat{L} \arrow[d, "{(m_L,a)}"'] \arrow[dr, phantom, "(FPBC)"] & \widehat{K} \arrow[l, "{(l,id_A)}"'] \arrow[r, "{(r,id_A)}"] \arrow[d, "{(m_K,a)}"] \arrow[dr, phantom, "(PO)"] & \widehat{R} \arrow[d, "{(m_R,a)}"] \\
\widehat{G} & \widehat{D} \arrow[l, "{(l_1,id_{A_1})}"'] \arrow[r, "{(r_1,id_{A_1})}"] & \widehat{H}
\end{tikzcd}$$

SqPO-Rewriting of Attributed Structures

Attributed Structures:

$\widehat{\Delta}$:

$$
\begin{tikzcd}
\widehat{L} \arrow[d, "{(m_L,a)}"] \arrow[dr, phantom, "(FPBC)"] & \widehat{K} \arrow[l, "{(l,id_A)}"'] \arrow[r, "{(r,id_A)}"] \arrow[d, "{(m_K,a)}"] \arrow[dr, phantom, "(PO)"] & \widehat{R} \arrow[d, "{(m_R,a)}"] \\
\widehat{G} & \widehat{D} \arrow[l, "{(l_1,id_{A_1})}"'] \arrow[r, "{(r_1,id_{A_1})}"] & \widehat{H}
\end{tikzcd}
$$

x is context: *keep attribute*

$$l_1(x):t_1 \longleftarrow x:t_1 \longmapsto r_1(x):t_1$$

x is preserved by the rule:

$$
\begin{tikzcd}
l(x):t \arrow[d, mapsto] & x:t \arrow[l] \arrow[r, mapsto] \arrow[d, mapsto] & r(x):t \arrow[d, mapsto] \\
l_1(x):a(t) & x:a(t) \arrow[l] \arrow[r, mapsto] & r_1(x):a(t)
\end{tikzcd}
$$

keep attribute

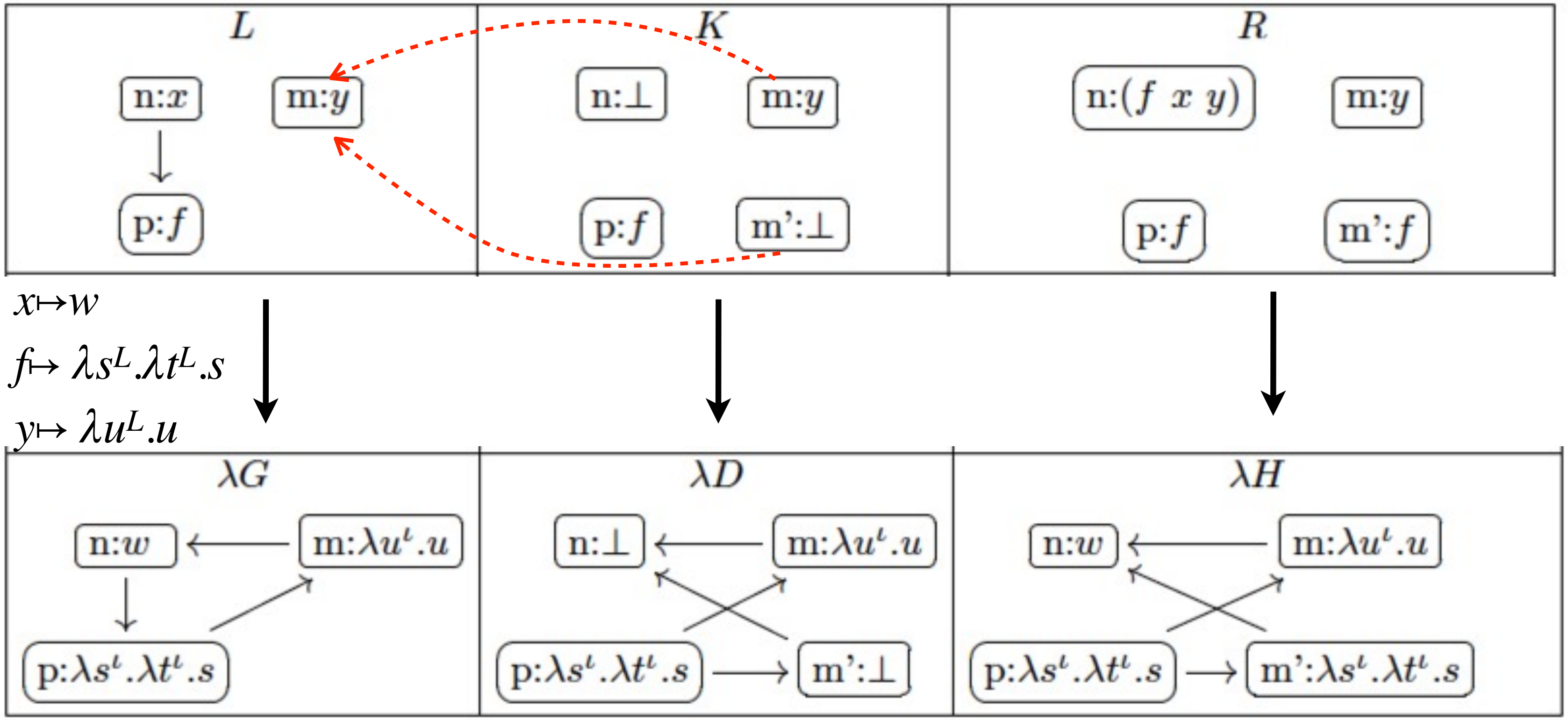
$$
\begin{tikzcd}
l()x:t \arrow[d, mapsto] & x:\bot \arrow[l] \arrow[r, mapsto] \arrow[d, mapsto] & r(x):t' \arrow[d, mapsto] \\
l_1(x):a(t) & x:\bot \arrow[l] \arrow[r, mapsto] & r_1(x):a(t')
\end{tikzcd}
$$

change attribute

SqPO-Rewriting of Attributed Structures

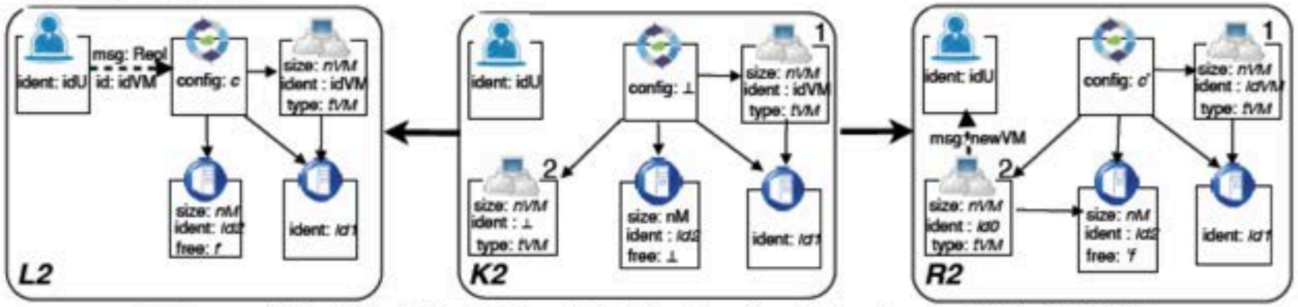
- A nice framework to define systems in the presence of cloning (and merging) operations
- Simple attribute handling:
  - ➡ allowing to use different kinds of values;
  - ➡ enabling a modular approach to prove properties (due to the independency of the structure from the attributes)

λ-Terms as Attributes



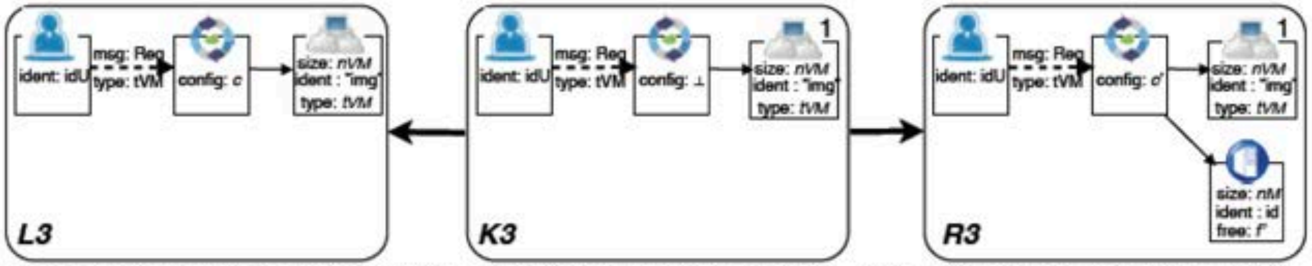
Cloud Administration

rule ReplicateVM



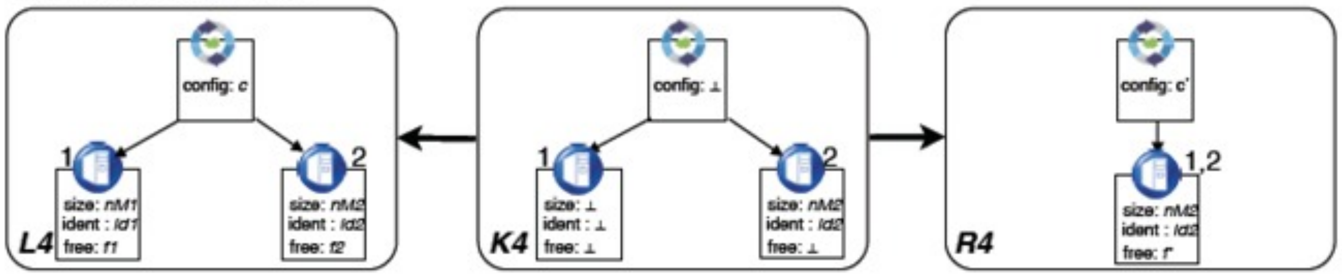
eqns: *newId(c,id0) ; id1 ≠ id2; nVM ≤ f ; f' = f - nVM ; c'= replVM(c,idU,id0)*

rule TurnOnMachine



eqns: *not(enoughSpace(c,nVM)); newId(c,id) ; nVM ≤ nM ; f'= nM-nVM ; c'=newMch(c,id,nM,f')*

rule TurnOffMachine



eqns: *nM1-f1 ≤ f2; f' = f2 - (nM1-f1) ; c'= mergeMch(c,id1,id2)*

Future Work

- Analysis of SqPO-transformation systems over attributed structures
- Case studies
- Tool support

Transformations of Attributed Structures with Cloning

Thanks for your attention!

Dominique Duval, Rachid Echahed, Frederic Prost, Leila Ribeiro

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

LIG

.inf INSTITUTO DE INFORMÁTICA UFRGS