Specification of a legacy tool by means of a dependency graph to improve its reusability

Paola Vallejo
Mickaël Kerboeuf
Jean-Philippe Babau

{vallejoco, kerboeuf, babau}@univ-brest.fr

University of Brest, France
Lab-STICC

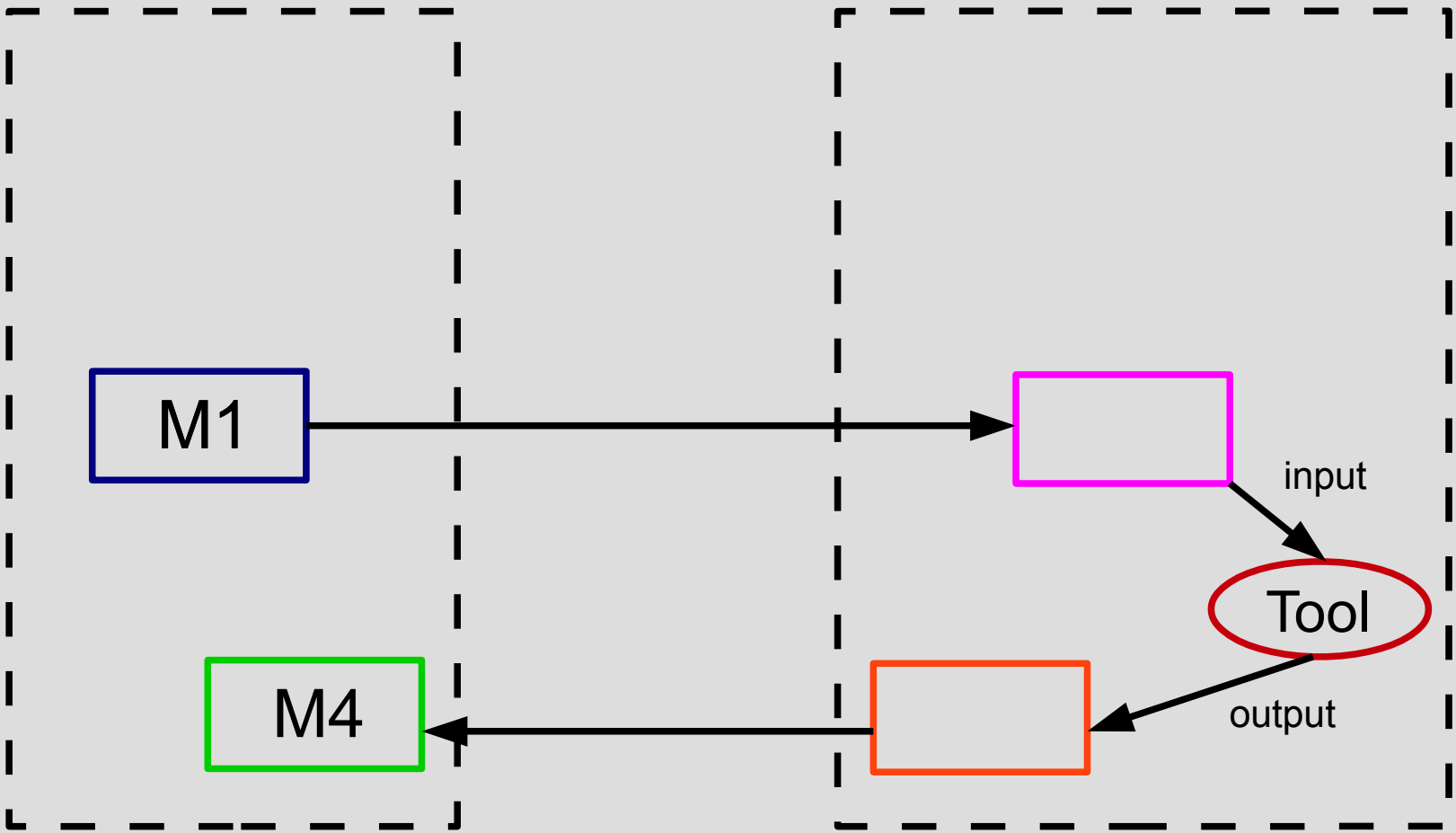
30th September 2013

Context

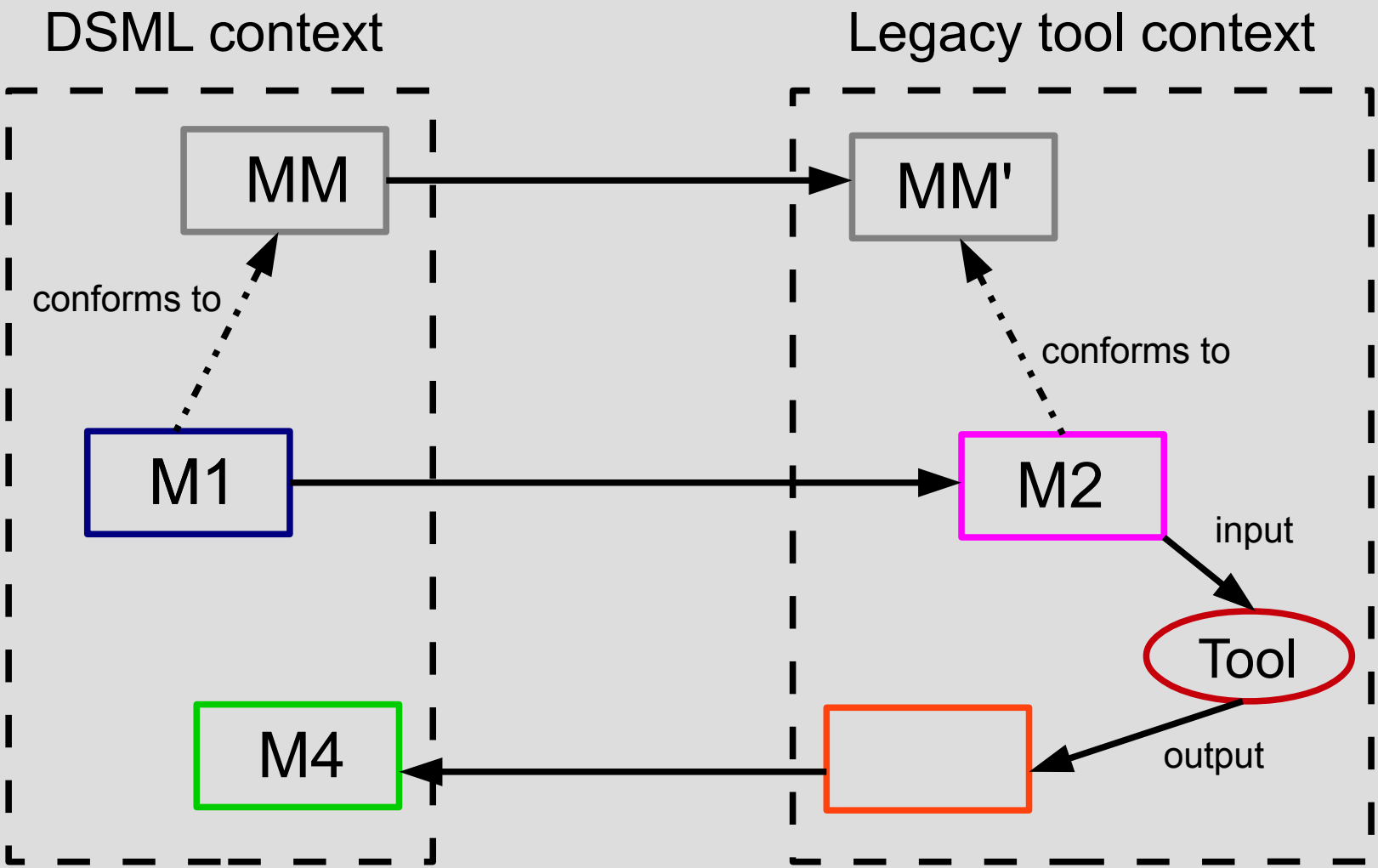
- MDE (Model Driven Engineering)
- DSML (Domain Specific Modeling Languages)

- Promptly making available tool support
- Reuse instead of rewrite

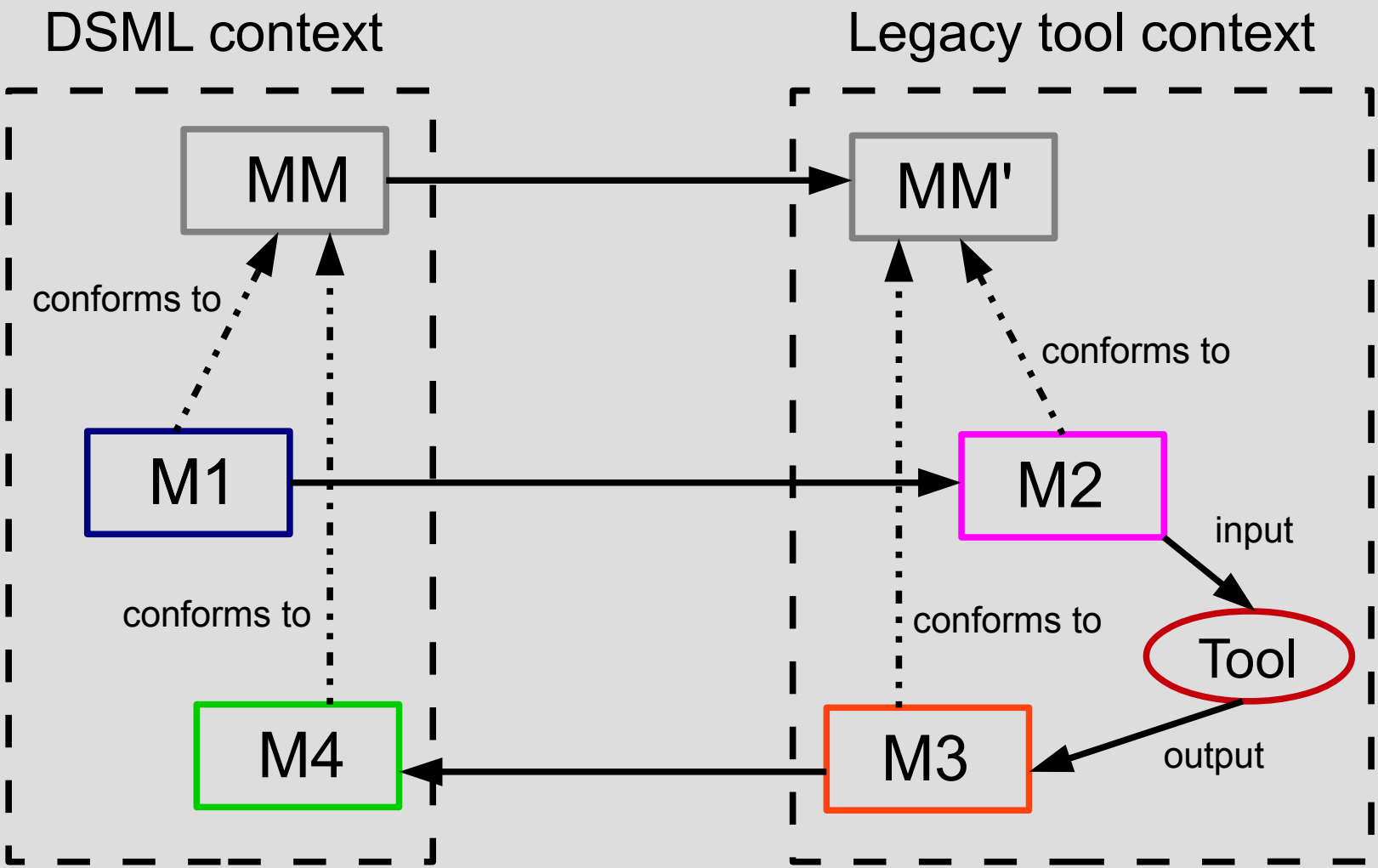
Context



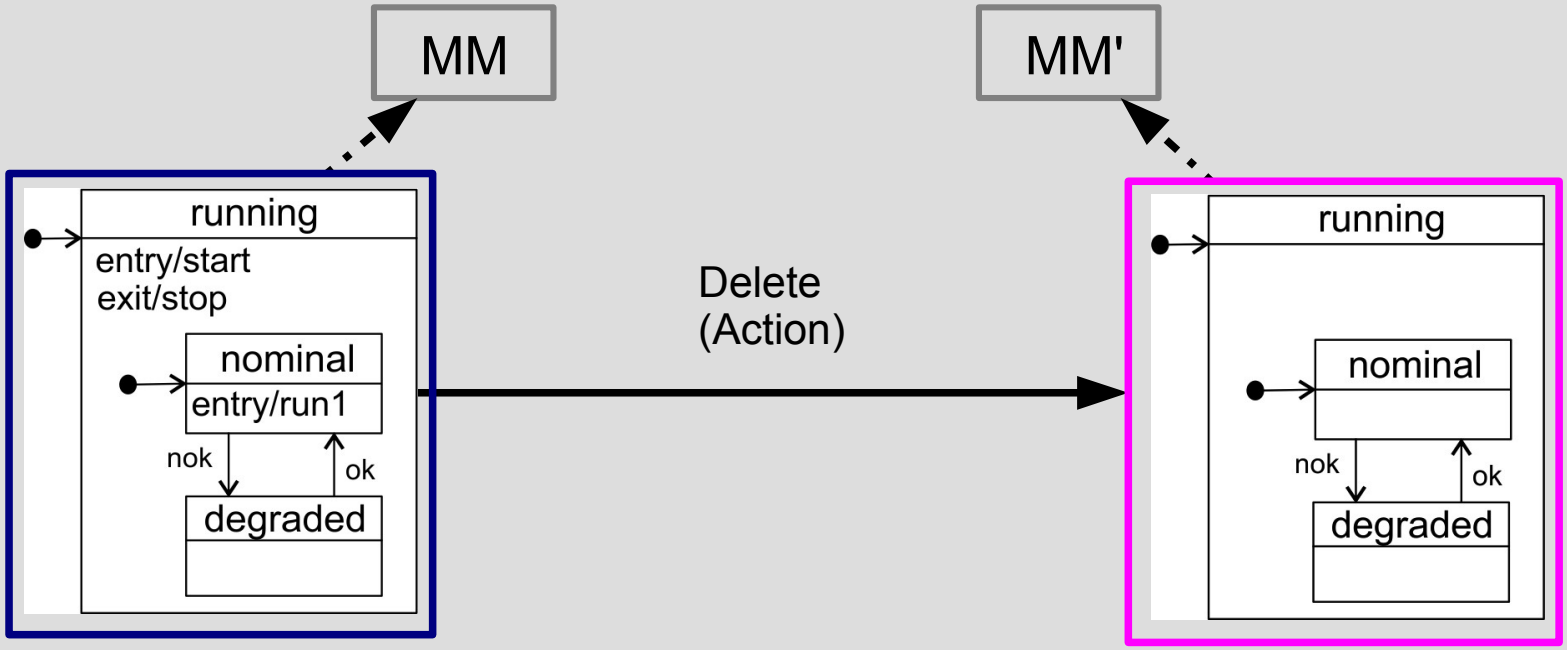
Context


DSML context
Legacy tool context
MM
MM'
conforms to
conforms to
M1
M2
input
Tool
M4
output

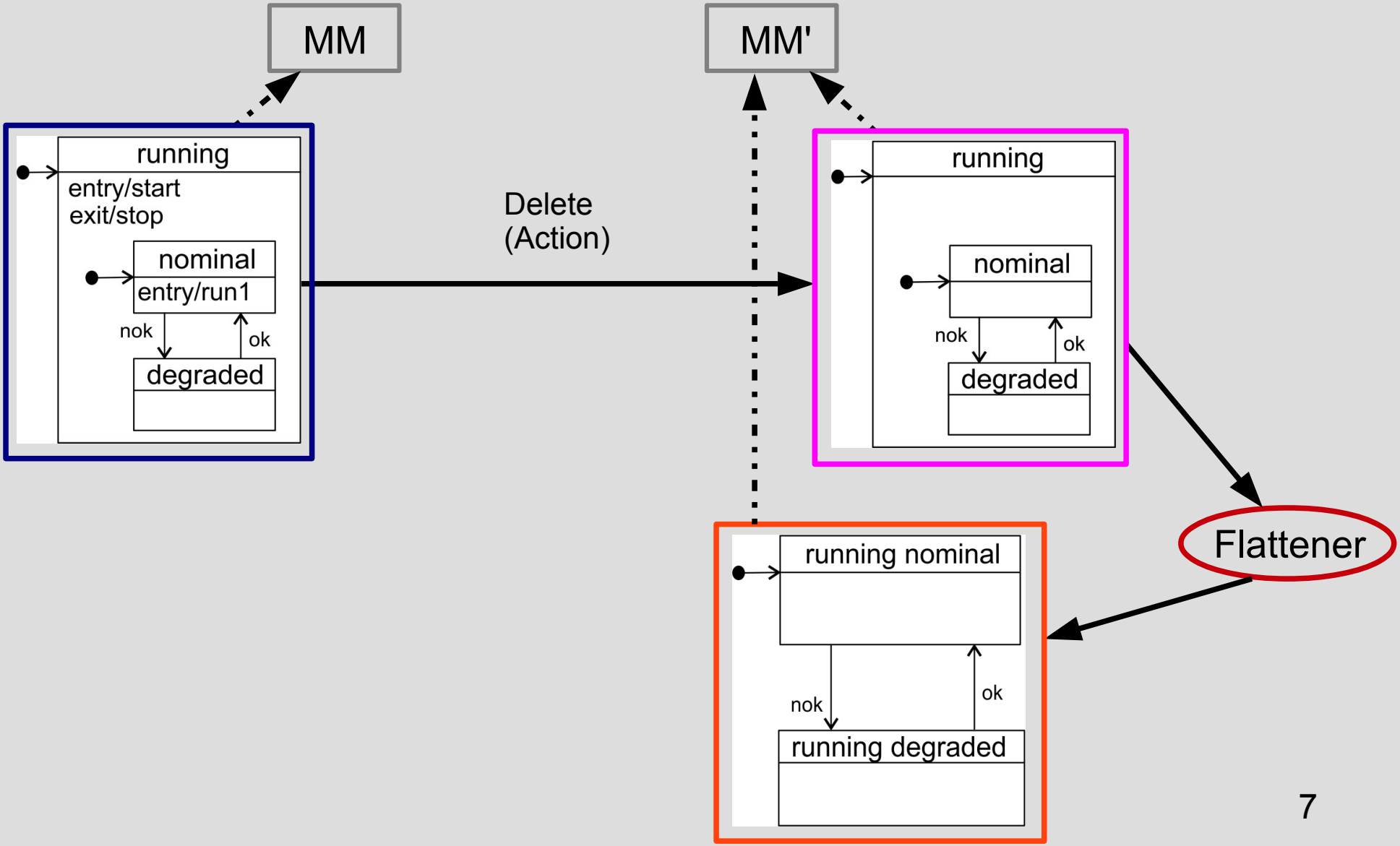
Context


DSML context
Legacy tool context
MM
MM'
conforms to
conforms to
M1
M2
input
conforms to
conforms to
Tool
M4
M3
output

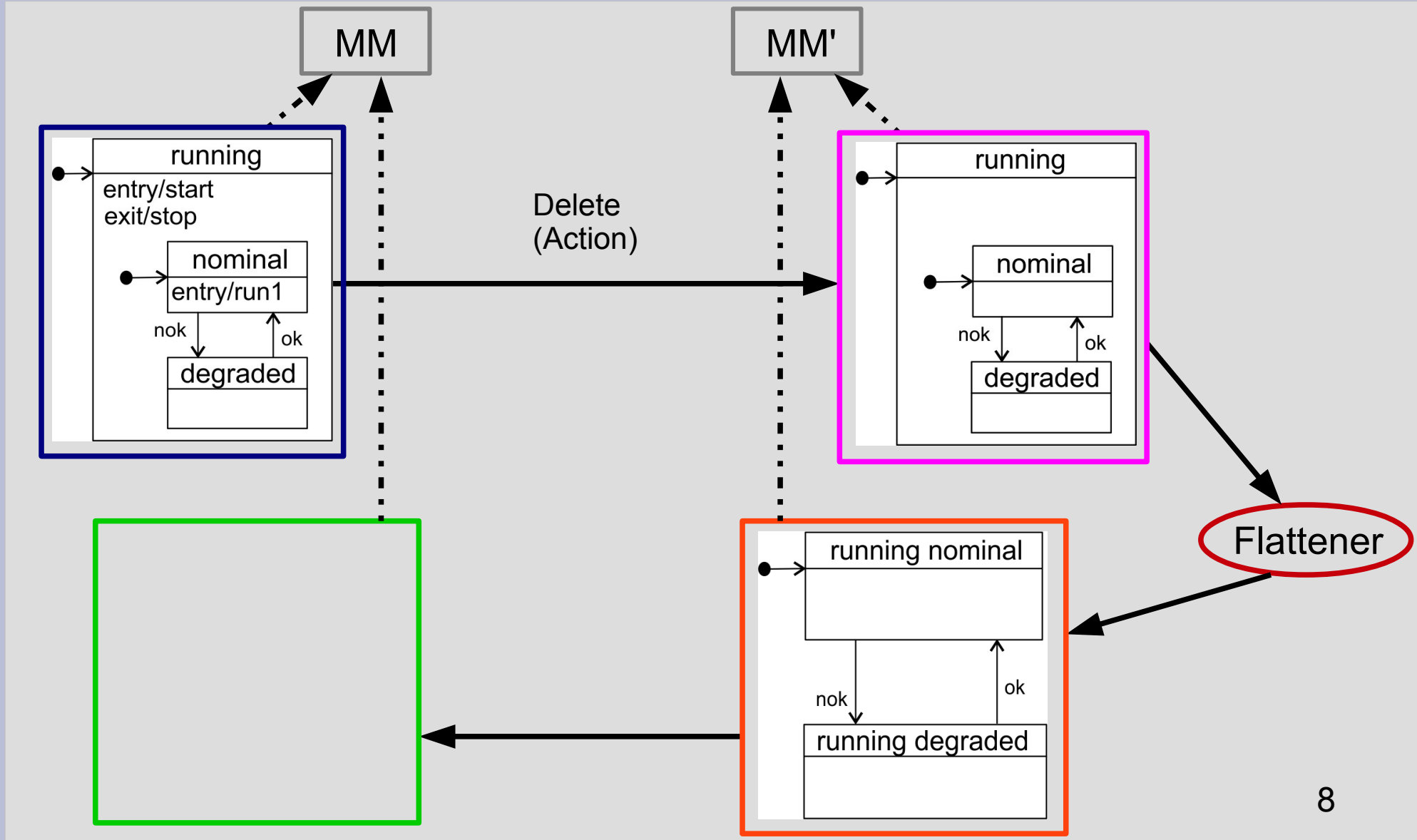
Example


MM
MM'
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
Delete
(Action)
running
nominal
nok
ok
degraded

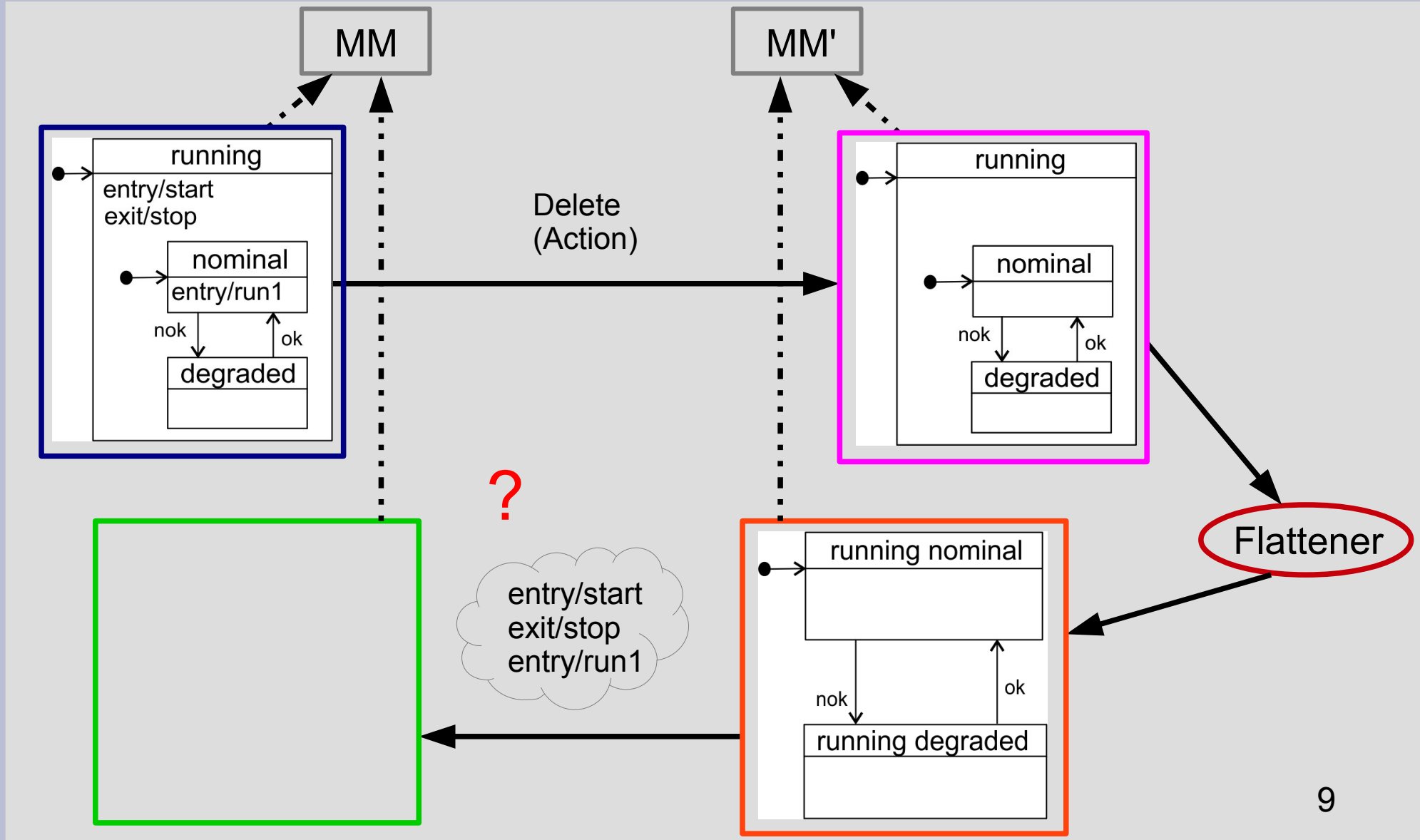
Example


MM
MM'
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
Delete
(Action)
running
nominal
nok
ok
degraded
Flattener
running nominal
nok
ok
running degraded

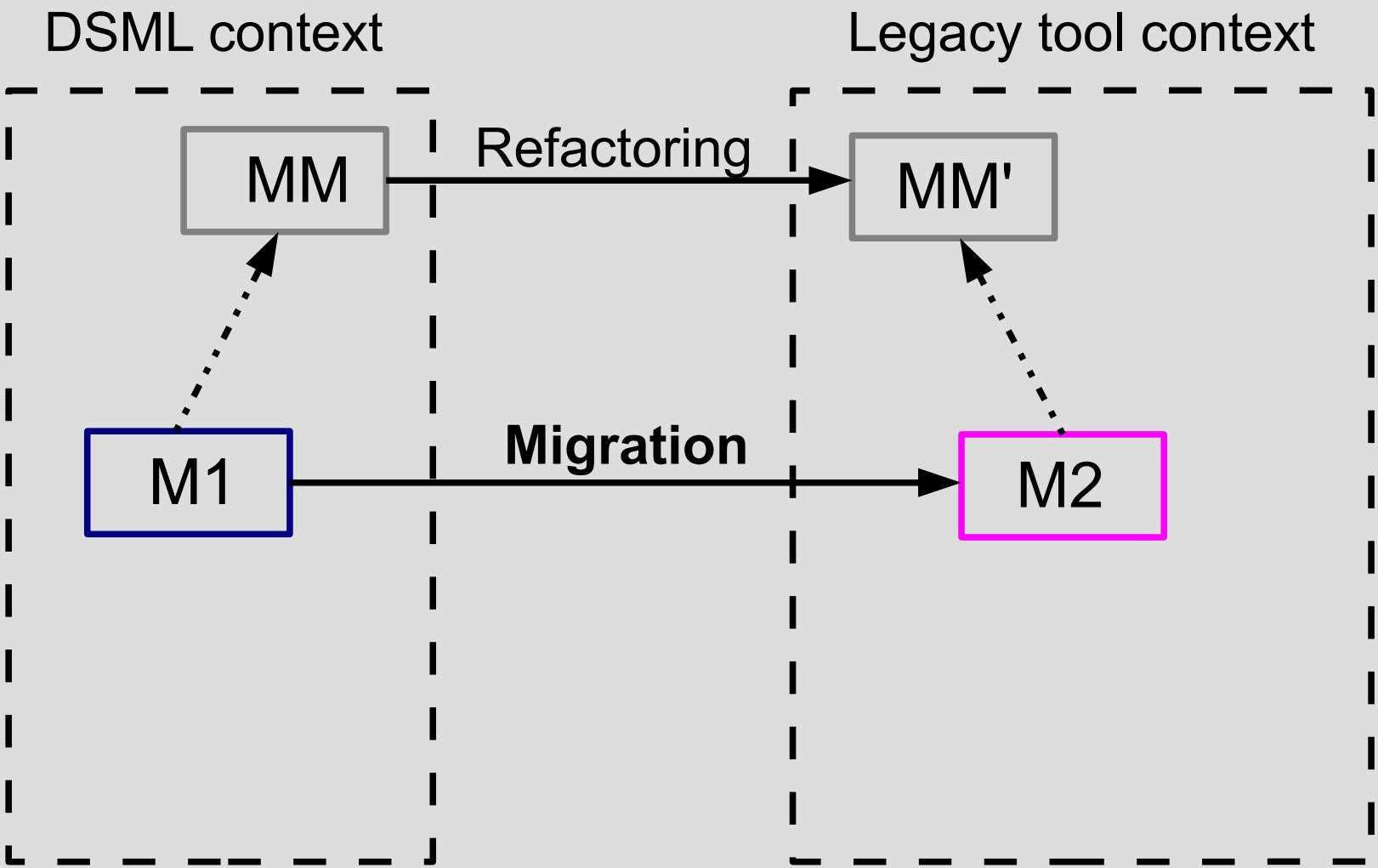
Example


MM
MM'
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
Delete
(Action)
running
nominal
nok
ok
degraded
Flattener
running nominal
nok
ok
running degraded

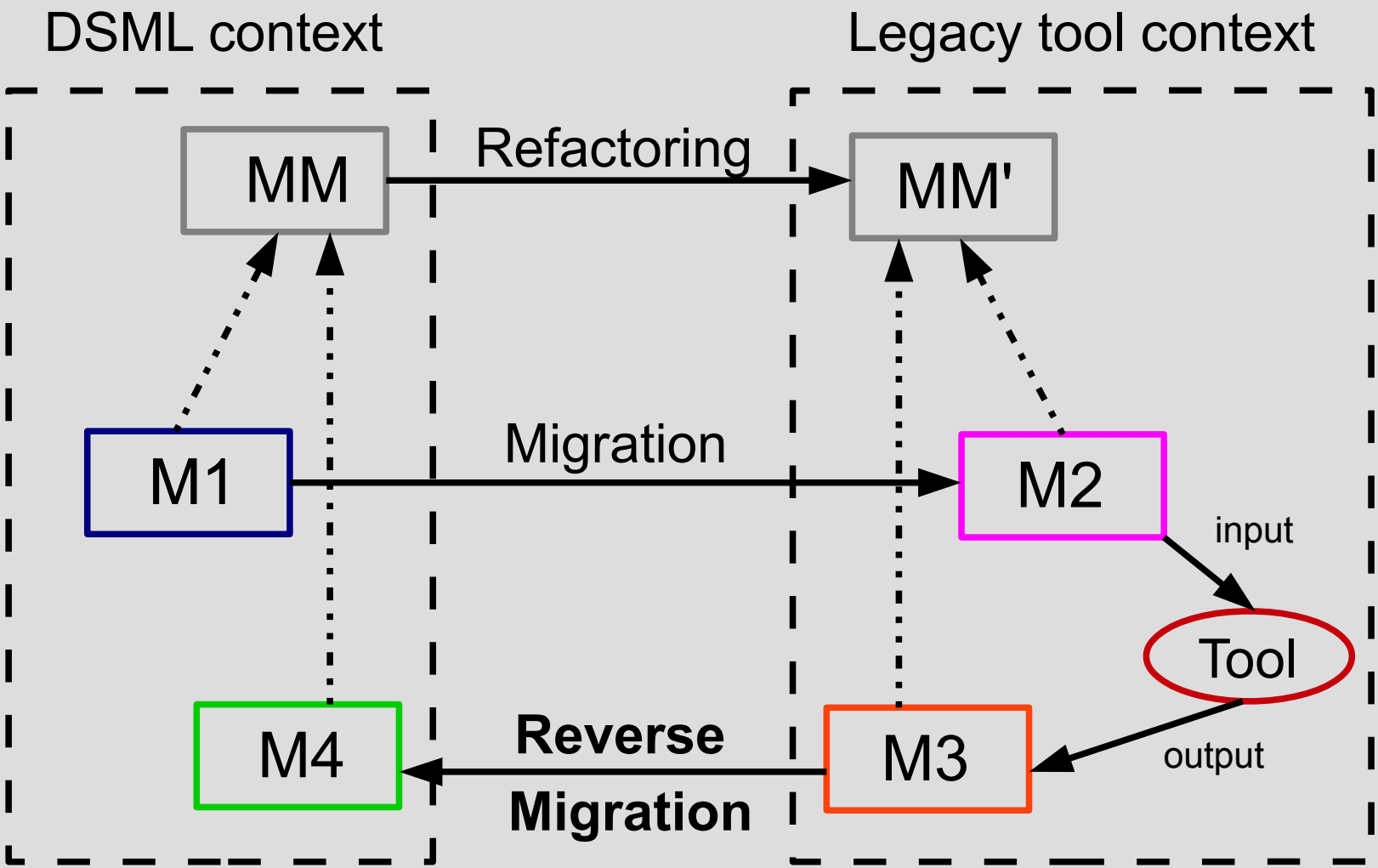
Example


MM
MM'
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
Delete
(Action)
running
nominal
nok
ok
degraded
Flattener
?
entry/start
exit/stop
entry/run1
running nominal
nok
ok
running degraded

Challenge


DSML context
Legacy tool context
MM
Refactoring
MM'
M1
Migration
M2

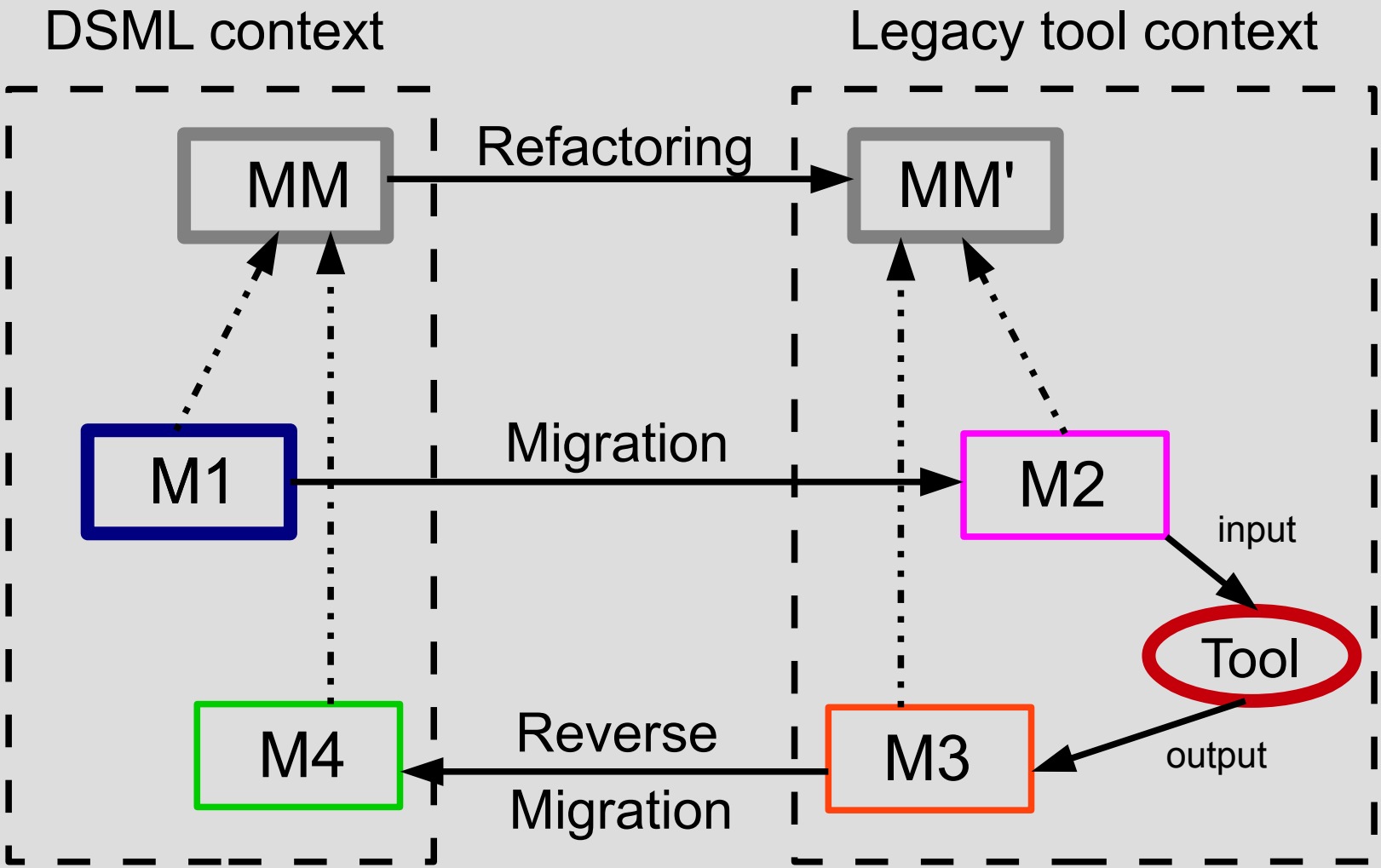
Challenge


DSML context
Legacy tool context
MM
Refactoring
MM'
M1
Migration
M2
input
Tool
M4
Reverse
Migration
M3
output

Agenda

- Context
- Example
- Challenge
- Key mechanism
- Dependency graph
- Implementation
- Conclusion

Modif Reuse

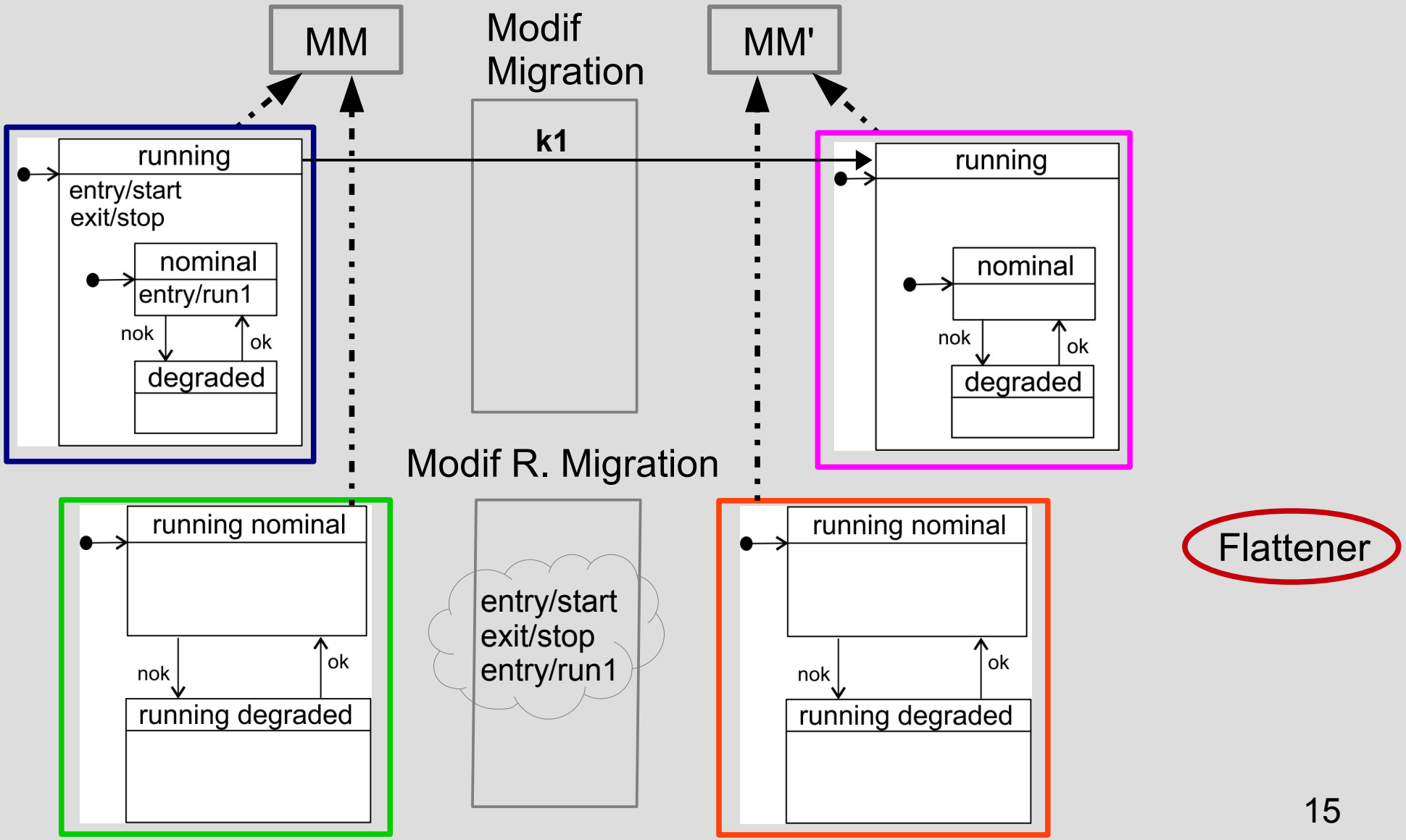

DSML context
Legacy tool context
MM
Refactoring
MM'
M1
Migration
M2
input
Tool
M4
Reverse
Migration
M3
output

Keys mechanism

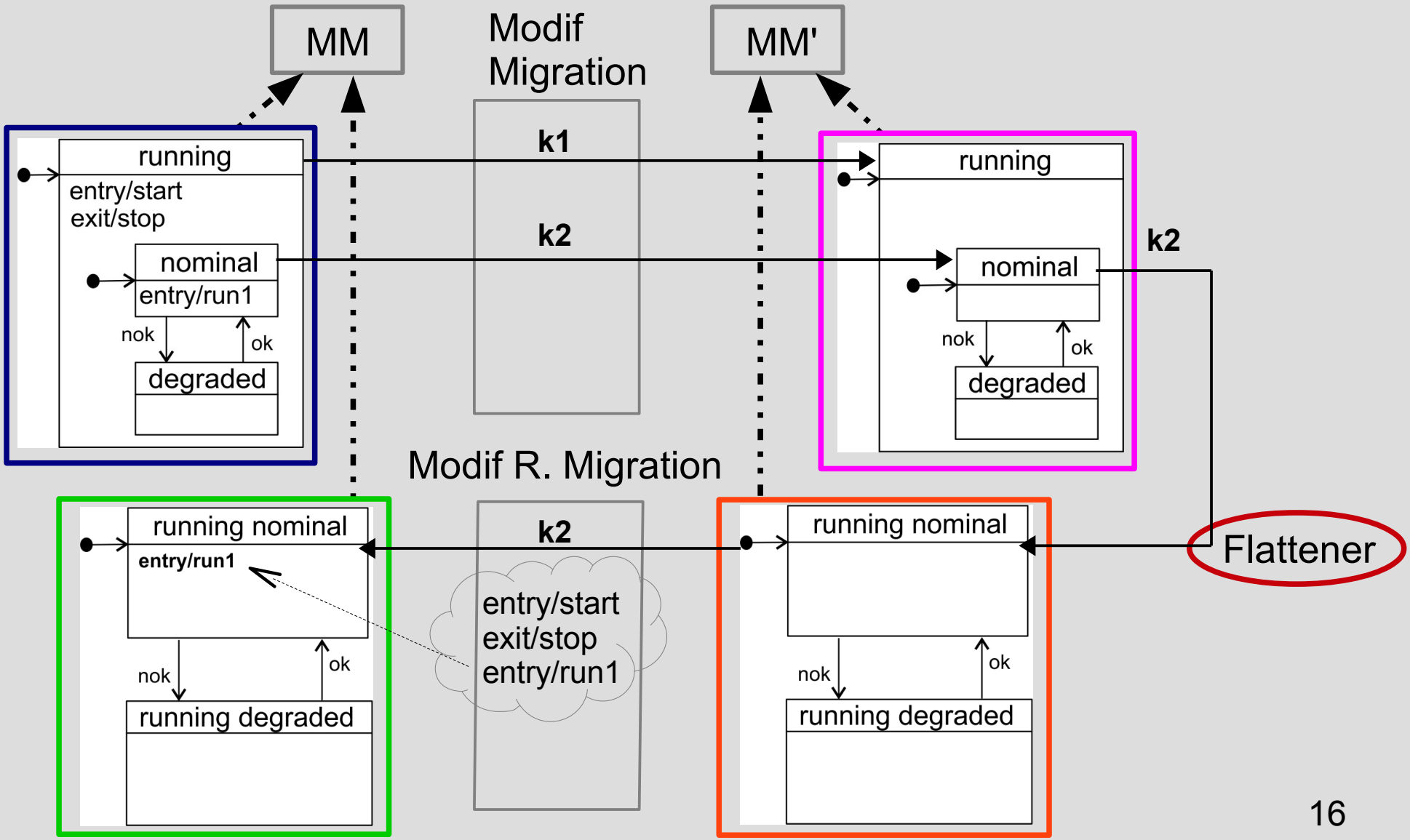
- Unique identification
- 1 source instance → 1 target instance
- Identifies in M2 and M3 deleted instances of M1
- Recover and contextualize deleted instances

- Loss of some instances
- New instances

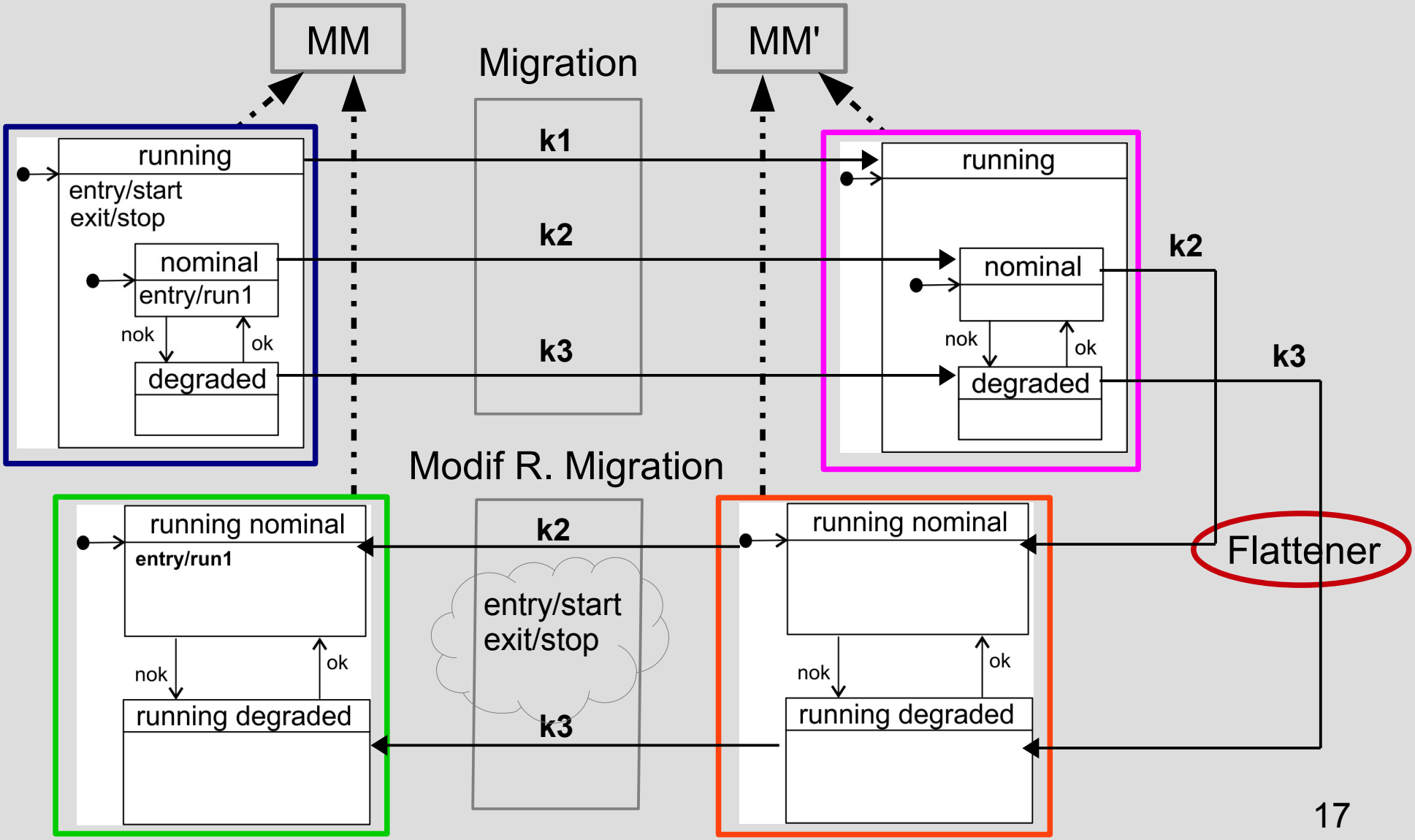
Example


MM
Modif
Migration
MM'
k1
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
Modif R. Migration
running nominal
nok
ok
running degraded
entry/start
exit/stop
entry/run1
running nominal
nok
ok
running degraded
Flattener

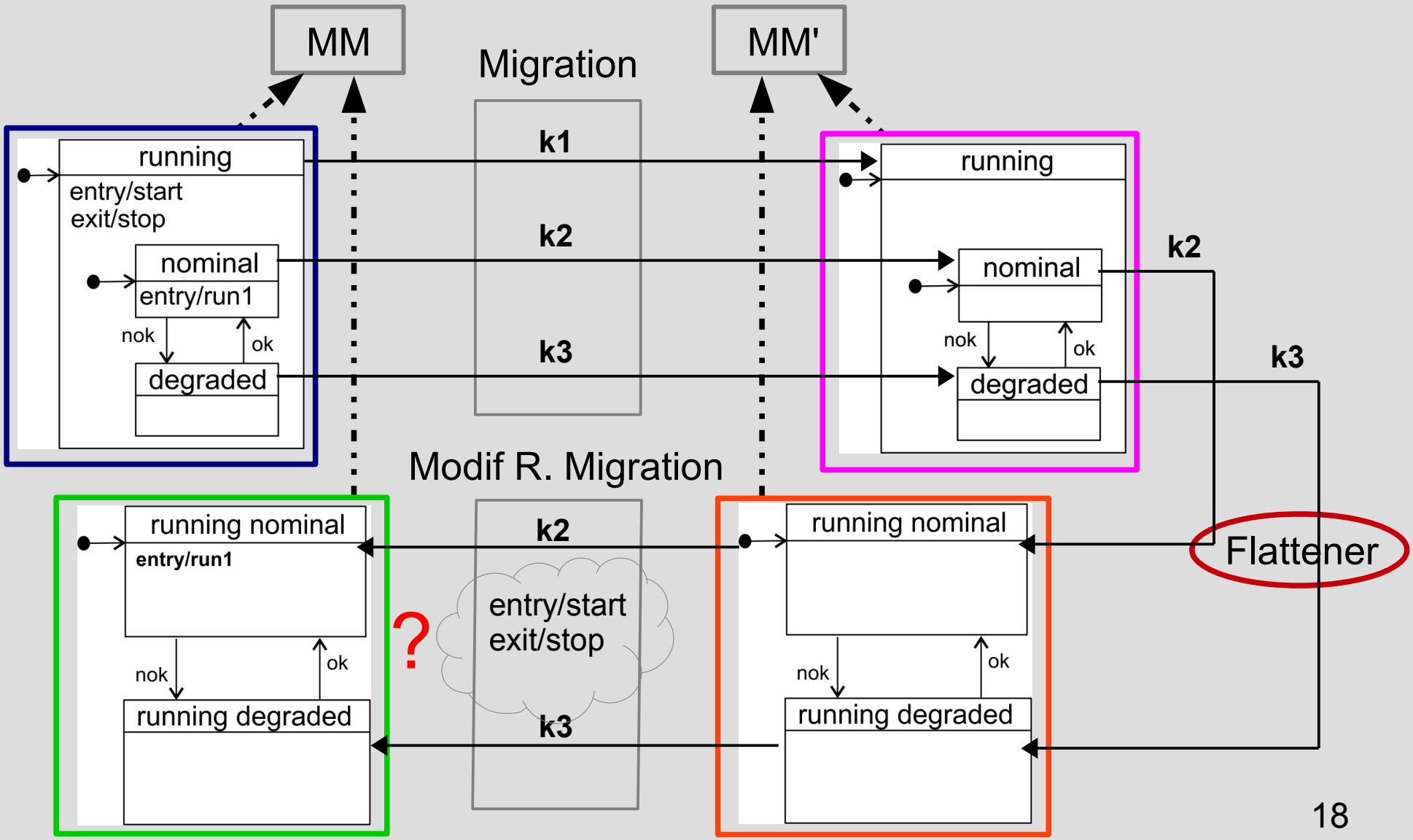
Example


MM
Modif Migration
MM'
k1
k2
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
k2
Modif R. Migration
k2
running nominal
entry/run1
nok
ok
running degraded
entry/start
exit/stop
entry/run1
running nominal
nok
ok
running degraded
Flattener

Example


MM
Migration
MM'
k1
k2
k3
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
k2
k3
Modif R. Migration
running nominal
entry/run1
nok
ok
running degraded
k2
entry/start
exit/stop
k3
running nominal
nok
ok
running degraded
Flattener

Example


MM
Migration
MM'
k1
k2
k3
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
k2
k3
Modif R. Migration
running nominal
entry/run1
nok
ok
running degraded
?
entry/start
exit/stop
Flattener

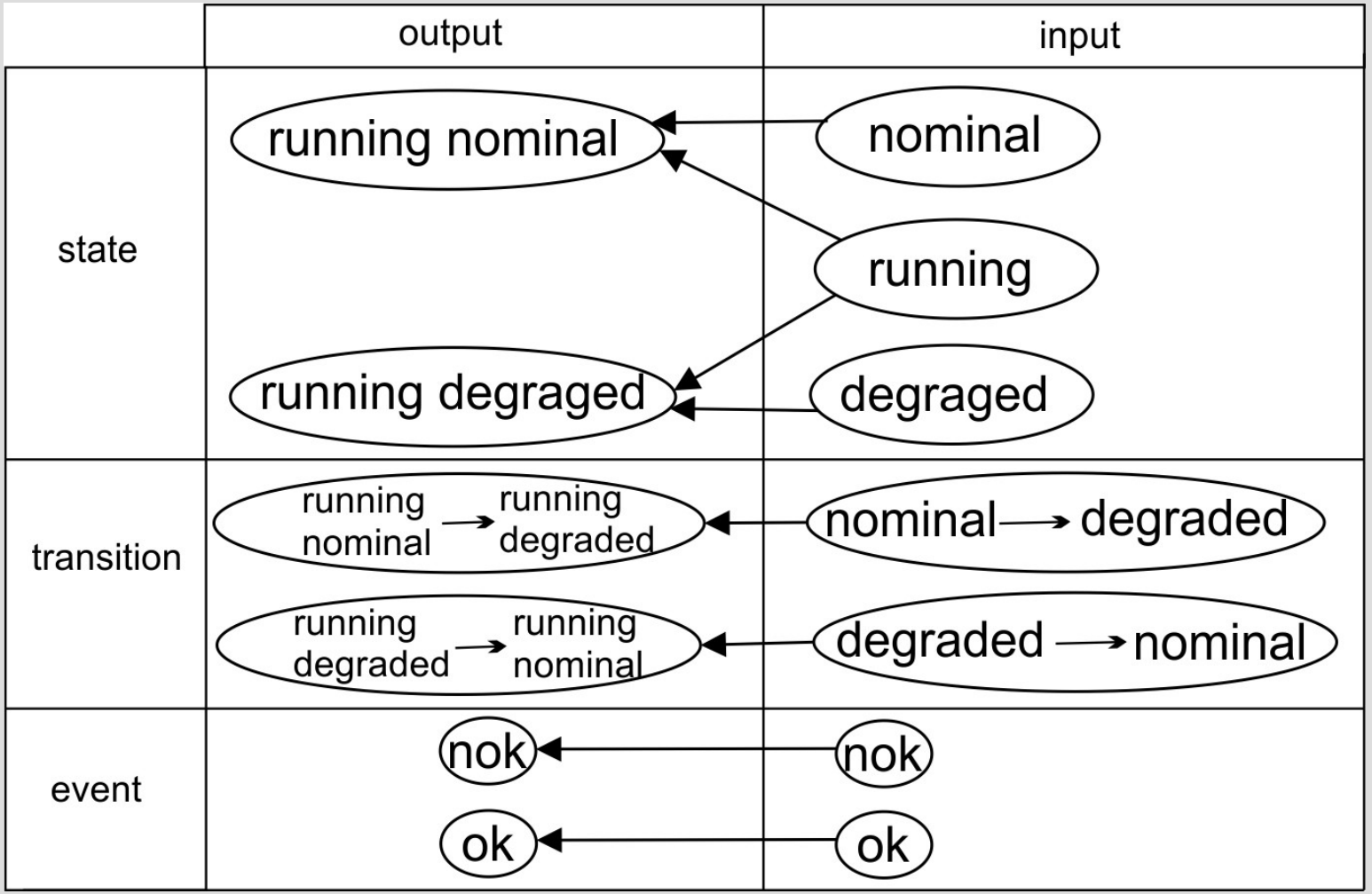
Agenda

- Context
- Example
- Challenge
- Key mechanism
- Dependency graph
- Implementation
- Conclusion

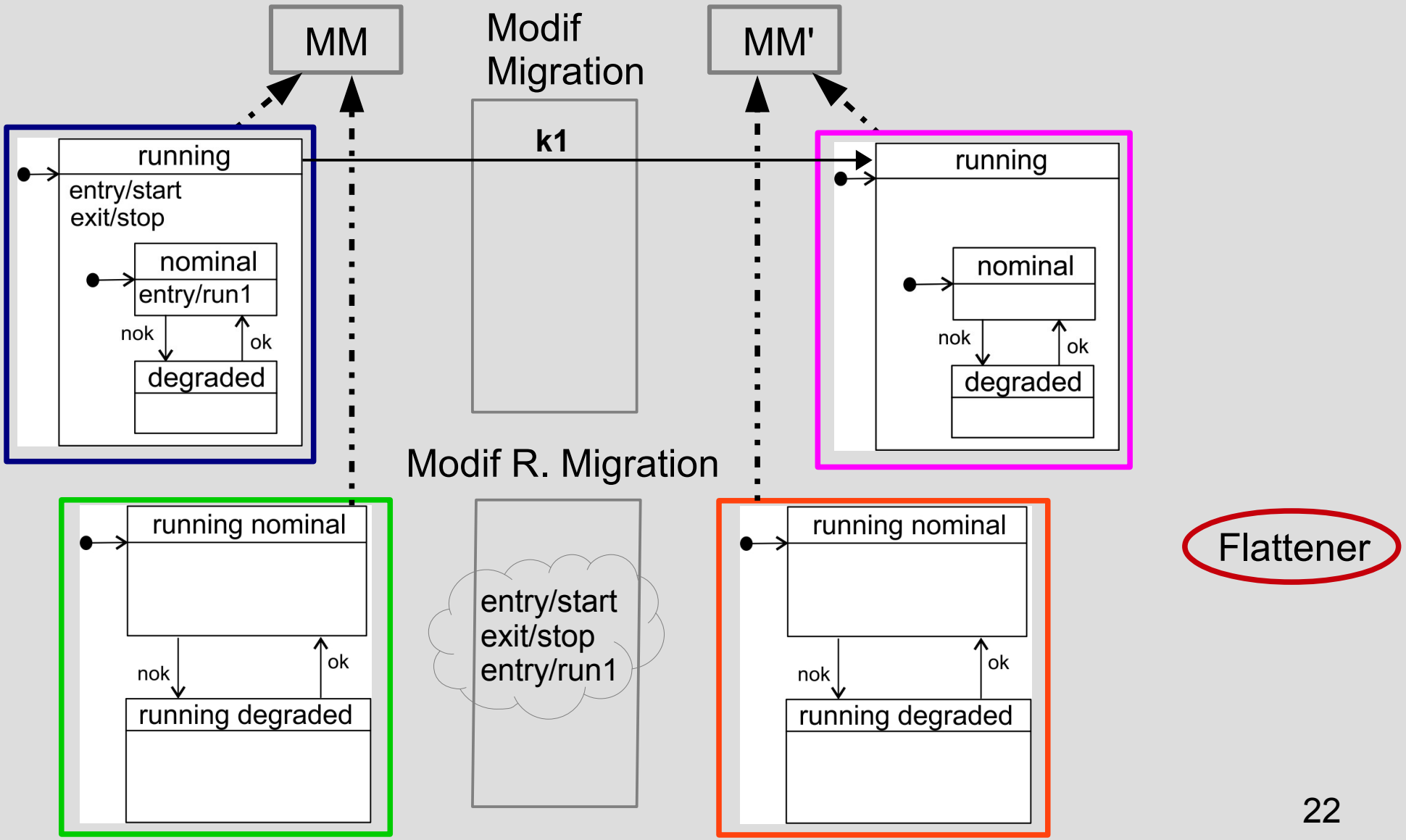
Dependency graph

- Specification of the legacy tool
- 1 output instance → 1 set of input instances
- Instances involved in modification or creation

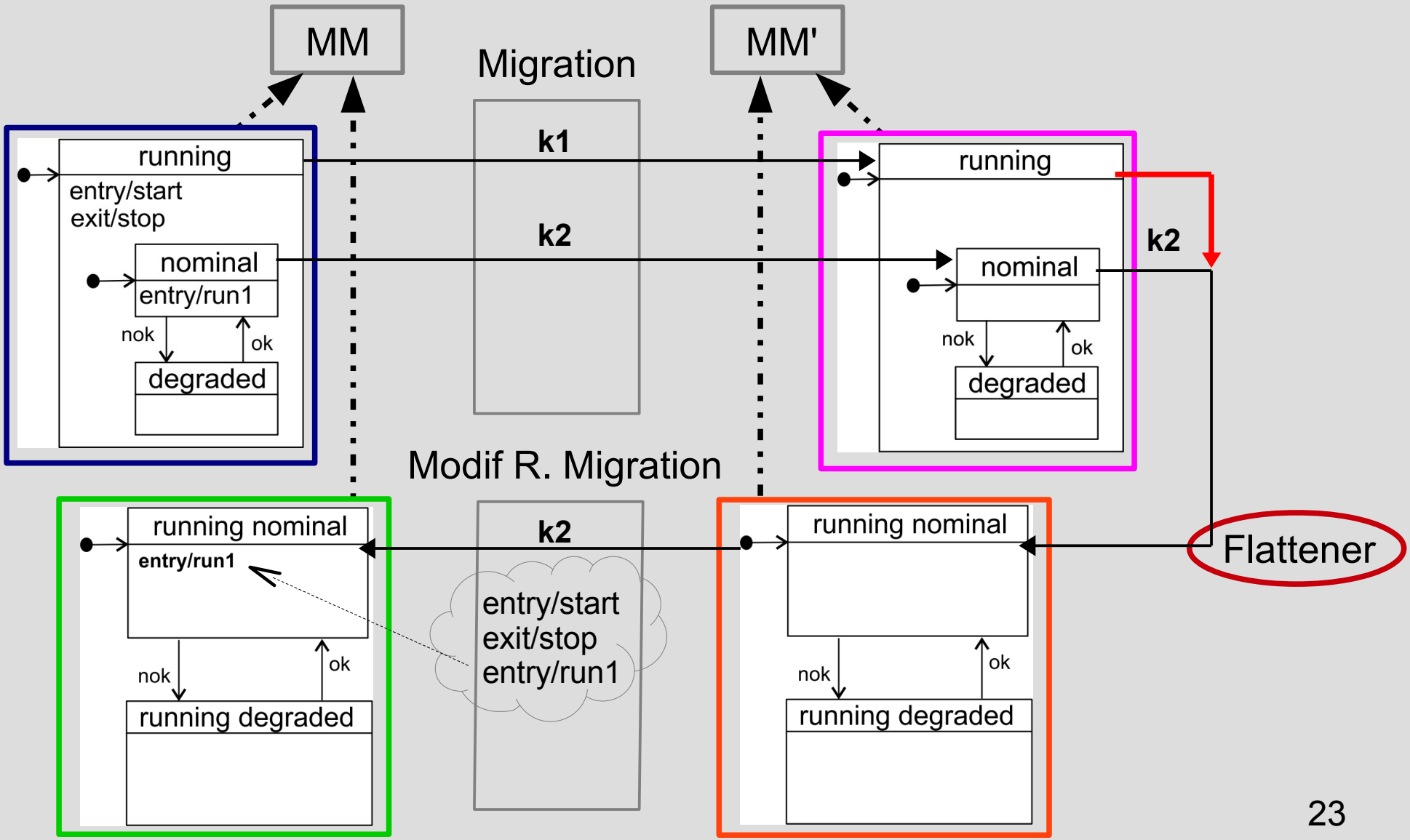
Dependency graph



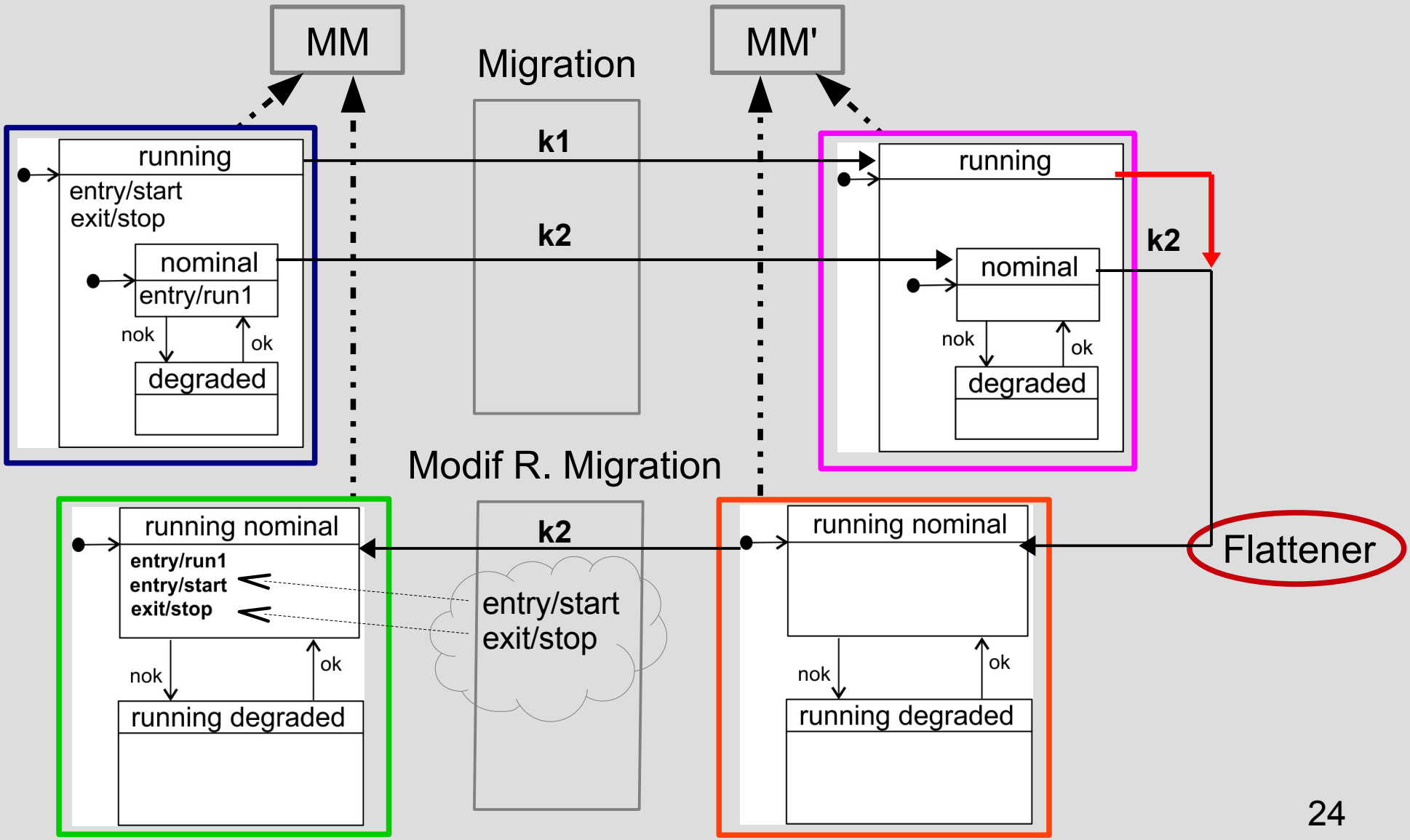
Example


MM
Modif Migration
MM'
k1
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
Modif R. Migration
running nominal
nok
ok
running degraded
entry/start
exit/stop
entry/run1
running nominal
nok
ok
running degraded
Flattener

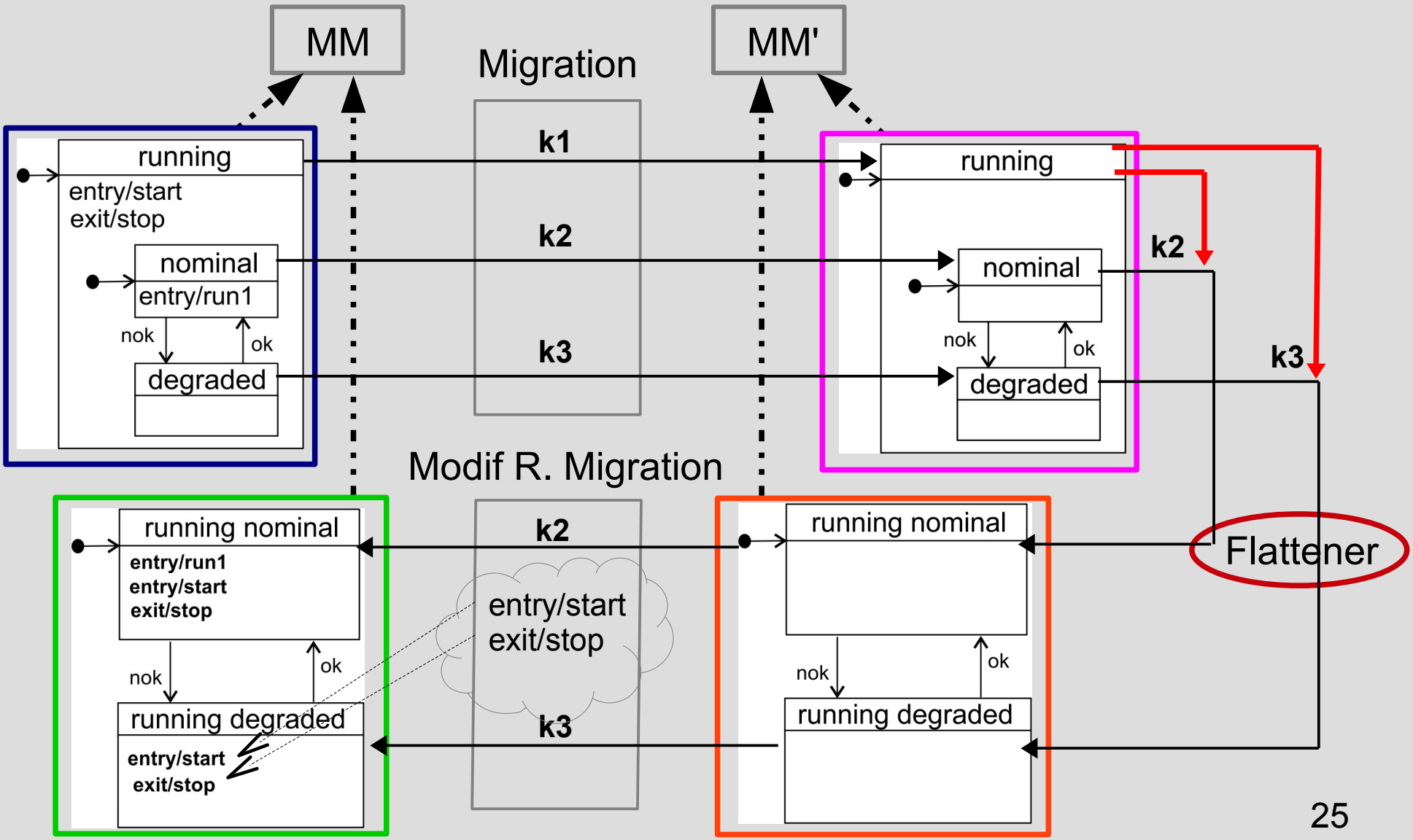
Example


MM
Migration
MM'
k1
k2
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
k2
Modif R. Migration
k2
running nominal
entry/run1
nok
ok
running degraded
entry/start
exit/stop
entry/run1
running nominal
nok
ok
running degraded
Flattener

Example


MM
Migration
MM'
k1
k2
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
running
nominal
nok
ok
degraded
k2
Modif R. Migration
k2
running nominal
entry/run1
entry/start
exit/stop
nok
ok
running degraded
entry/start
exit/stop
running nominal
nok
ok
running degraded
Flattener

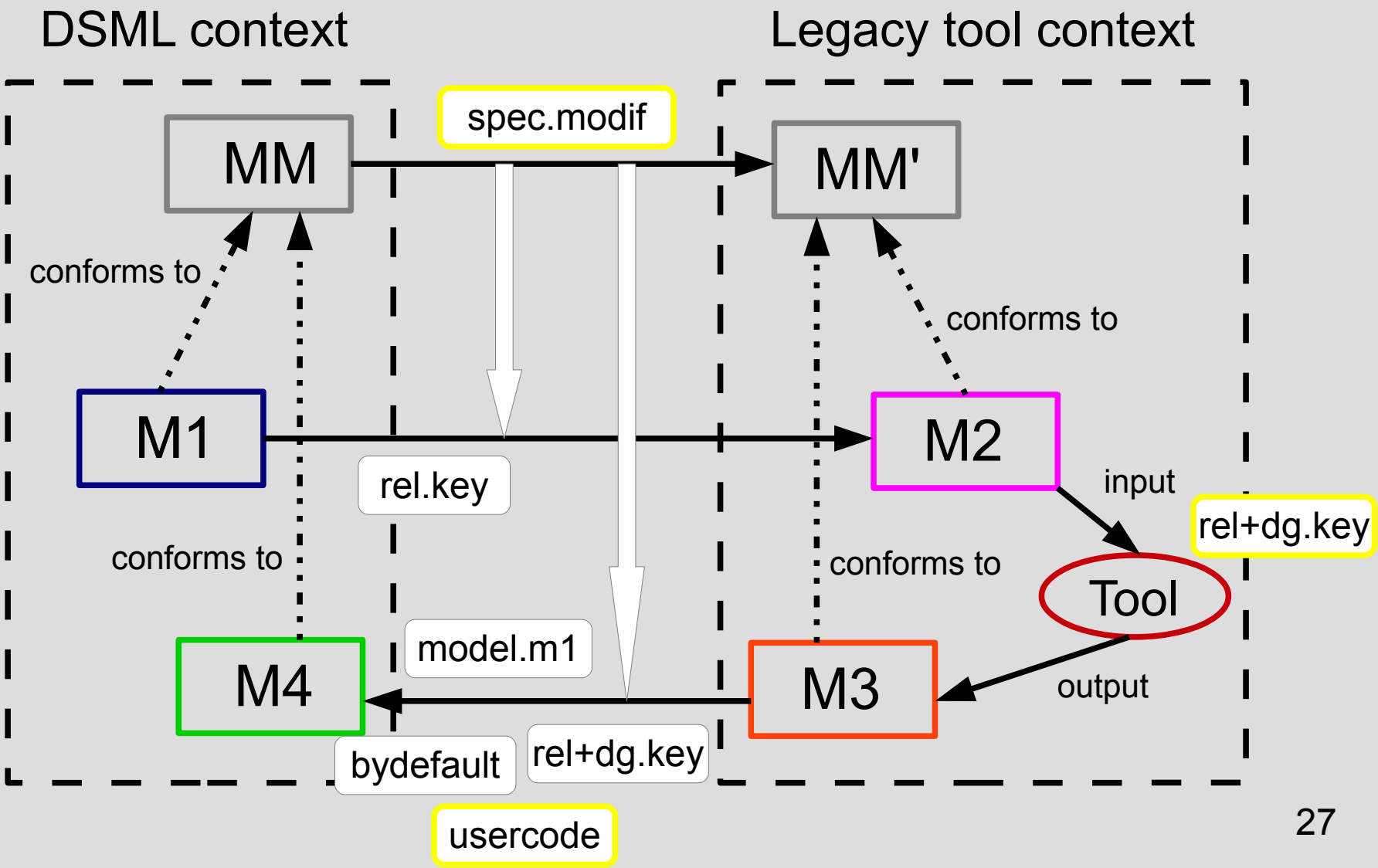
Example


MM
Migration
MM'
k1
k2
k3
running
entry/start
exit/stop
nominal
entry/run1
nok
ok
degraded
k2
k3
Modif R. Migration
running nominal
entry/run1
entry/start
exit/stop
nok
ok
running degraded
entry/start
exit/stop
Flattener

Agenda

- Context
- Example
- Challenge
- Key mechanism
- Dependency graph
- Implementation
- Conclusion

Implementation



User code example

```
public class DefaultBehavior implements Migration{

    public void putAction(M3.State state, Elist<M1.State> relatedStates){...}

    public void preFunction(){return ;}
    public void postFunction(){return ;}

    final void function(M3 M3model, Key dicoKeys, M1 M1model){
        for(M3.State state : M3StatesList){
            relatedStates = getRelatedStates(state);

            for(M1.State related : relatedStates){
                putAction(state, relatedStates);
            }
        }
    }
}
```

```
void migration (Migration byDefault){
    ...
    byDefault.preFunction();
    byDefault.function();
    byDefault.postFunction();
    ...
}
```

```
//execution of by default migration
migration (new DefaultBehavior()) ;
```

```
//execution of customized migration
migration (new CustomizedBehavior()) ;
```

```
public class CustomizedBehavior extends DefaultBehavior{
    ...
    public void preFunction(){...}
    public void putAction(M3.State state, Elist<M1.State> relatedStates){...}
    ...
}
```

Conclusion

- Additional information of the legacy tool
- Recover deleted instances
- Contextualize new instances
- Migration: metamodel dependent
- Reverse Migration: tool dependent

Questions ?

Thank you !