Splitting GSM Schemas: A Framework for Outsourcing of Declarative Artifact Systems

Rik Eshuis, Eindhoven University of Technology
Rick Hull, IBM T J Watson Research Center
Yutian Sun, UC Santa Barbara
Roman Vaculin, IBM T J Watson Research Center

IBM

TU/e Technische Universiteit Eindhoven University of Technology

Where innovation starts

Context

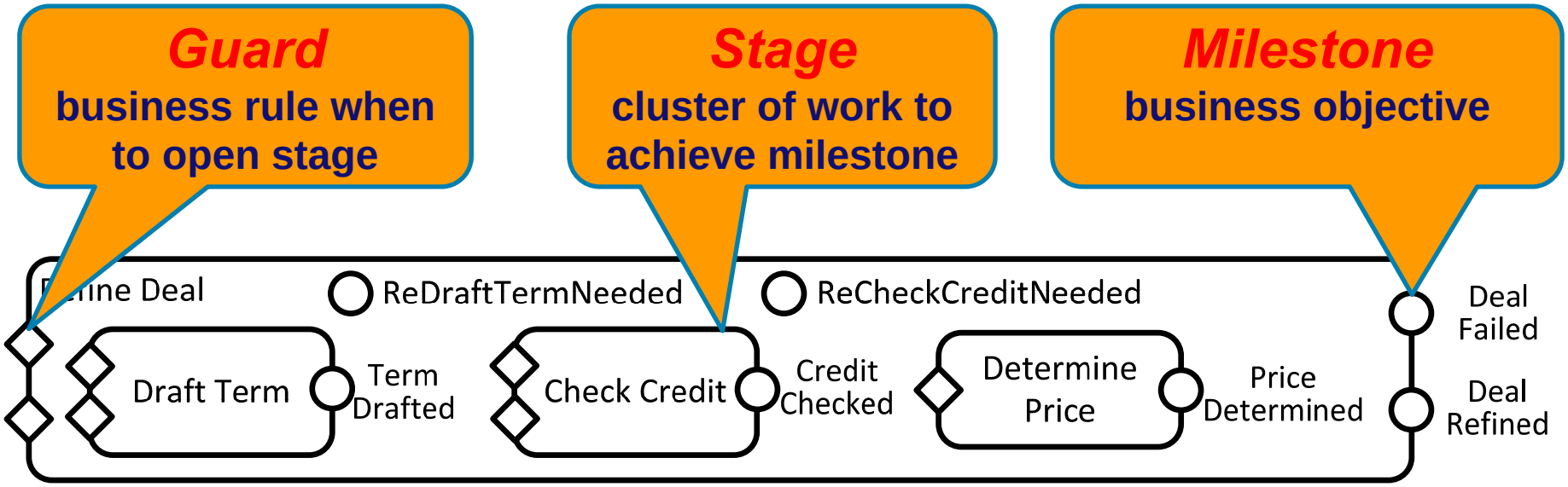
Motivation

- Business Process Outsourcing (BPO): part of a business process is performed by another organization
- Cloud-computing is key enabler of BPO

This study

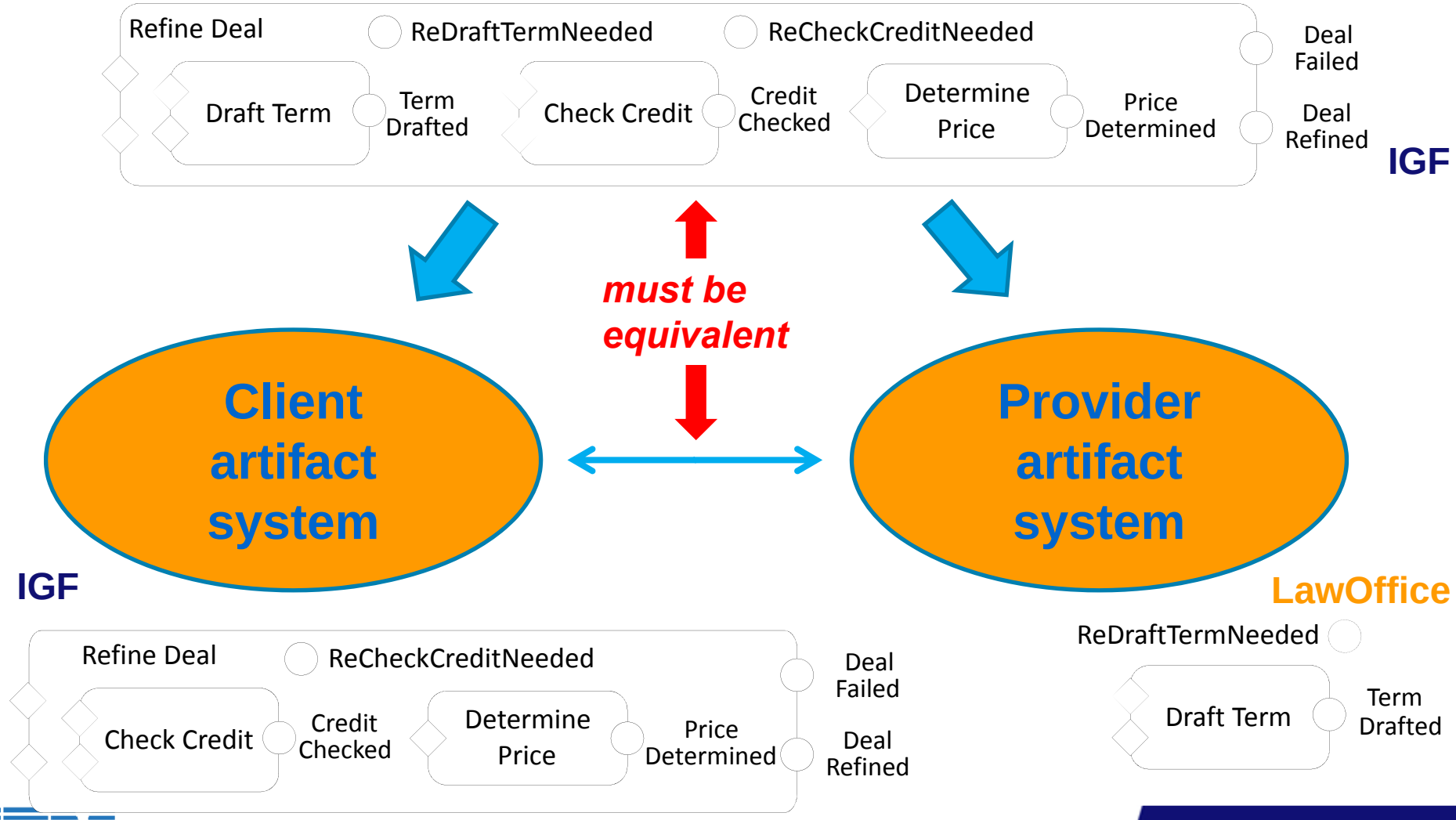
- BPO for declarative case management
  - Spez., Guard-Stage-Milestone business artifact model
  - GSM is basis for OMG's CMMN standard
- Main result: framework that
  - enables BPO at design-time and run-time
  - supports hiding business logic

Guard-Stage-Milestone (GSM) schemas



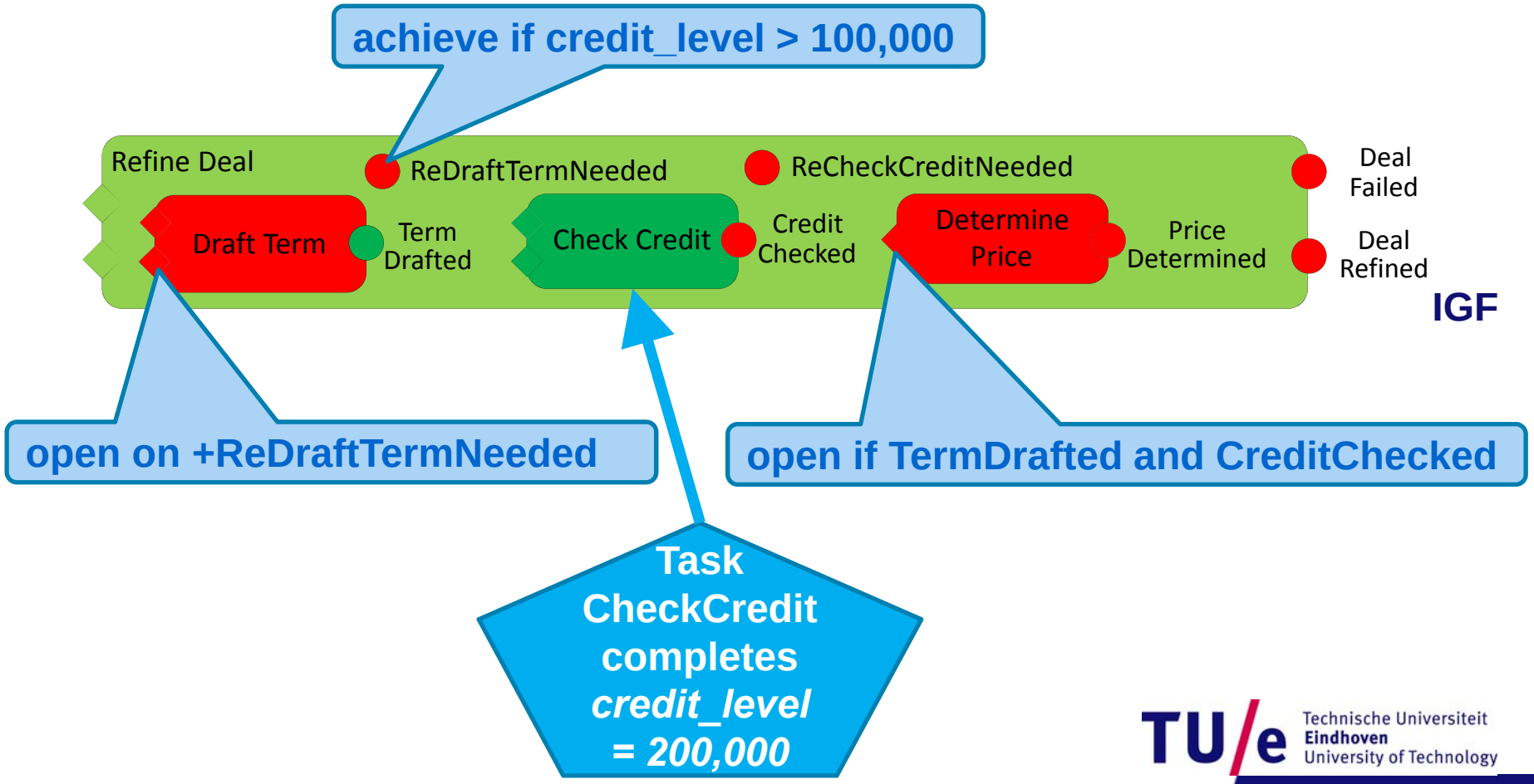
- Status of each stage,milestone ∈ { on, off}
- Business rules specify when stage/milestone changes state
  - Might refer to status of other stages/milestones
- Rules need to be evaluated in right order to ensure that all changes have maximal effect
- Unit of change triggered by external event is called a B-step

Problem: outsourcing GSM subschema



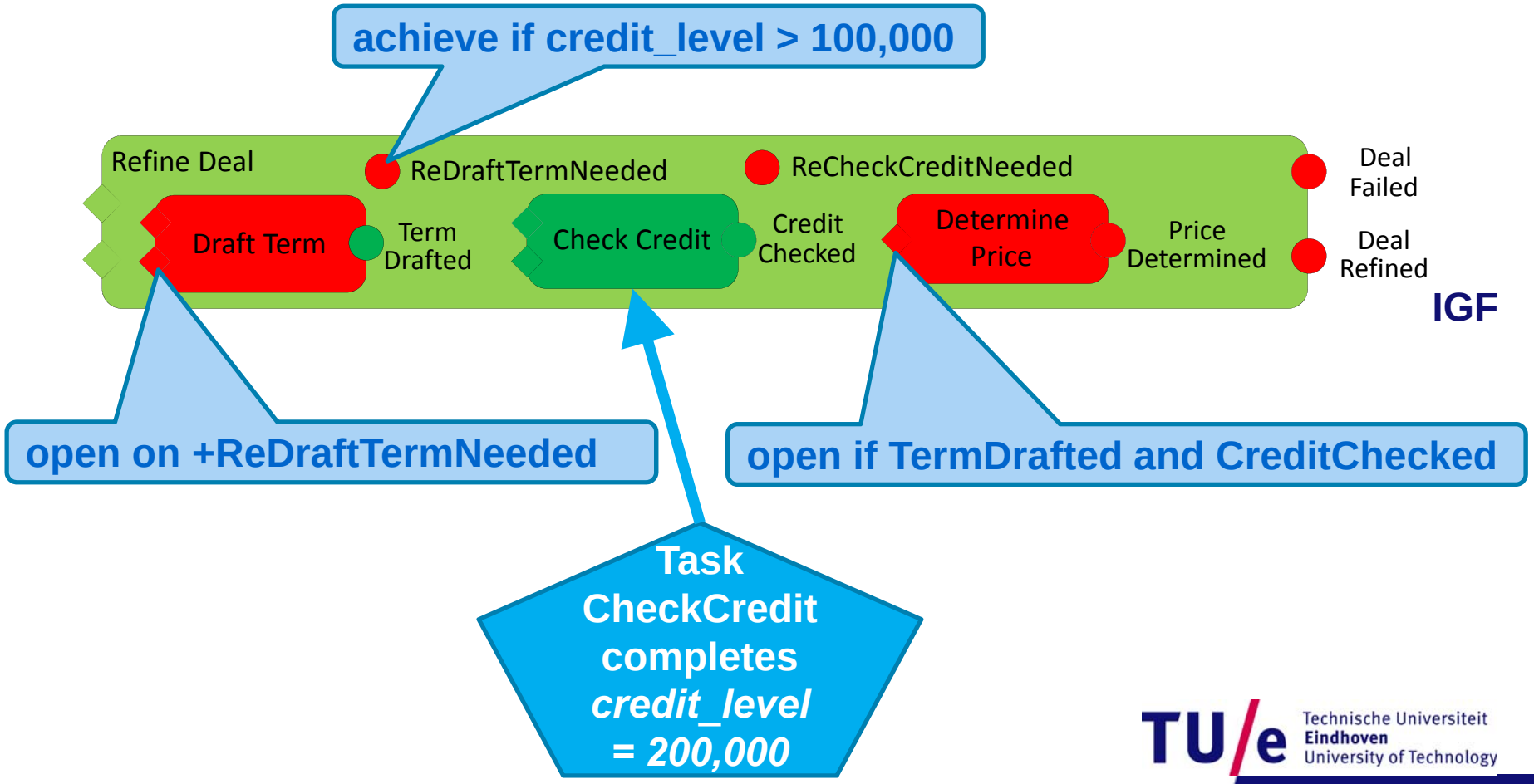
Example scenario in original GSM schema

- status is on
- status is off



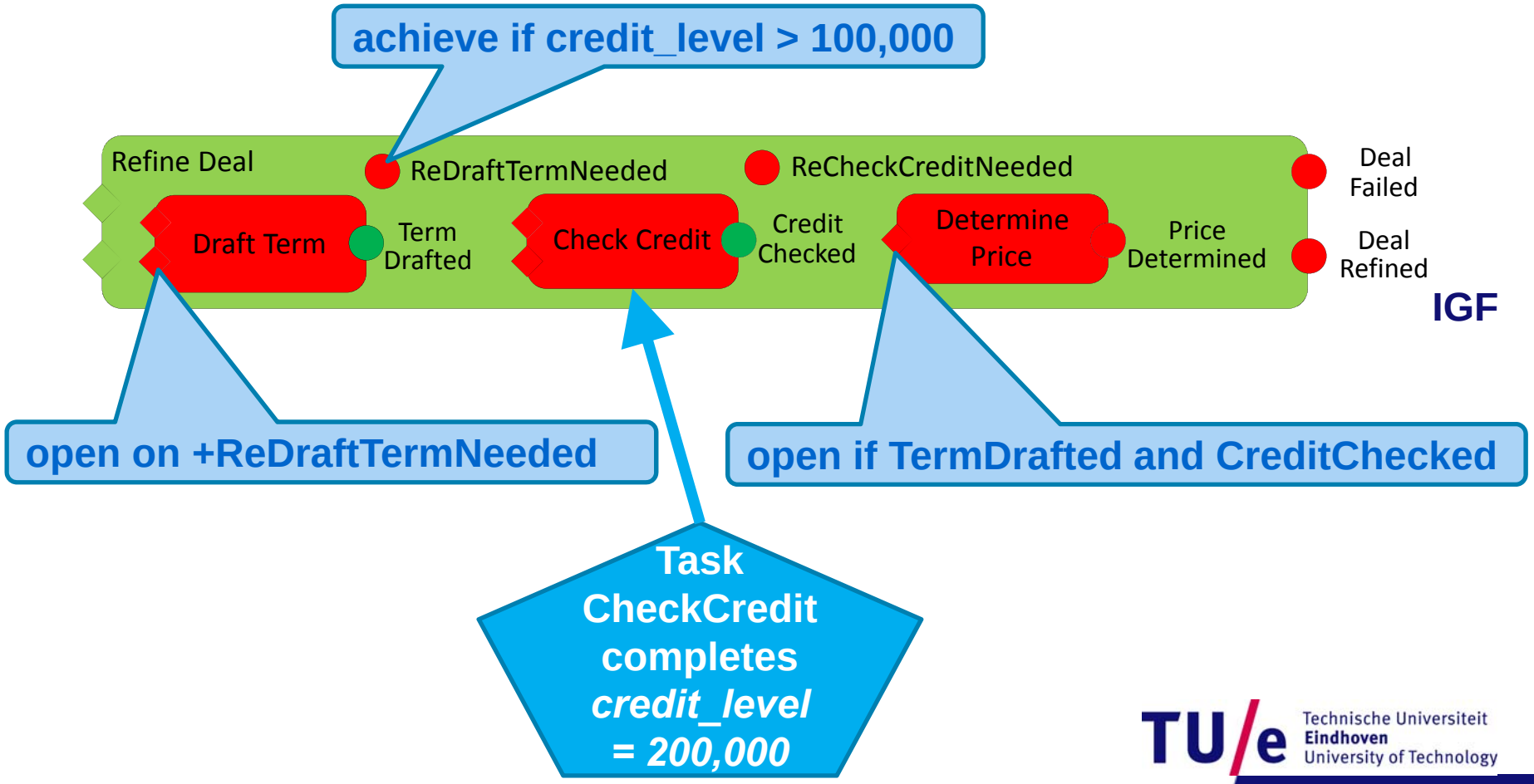
Starting B-step (1)

- status is on
- status is off



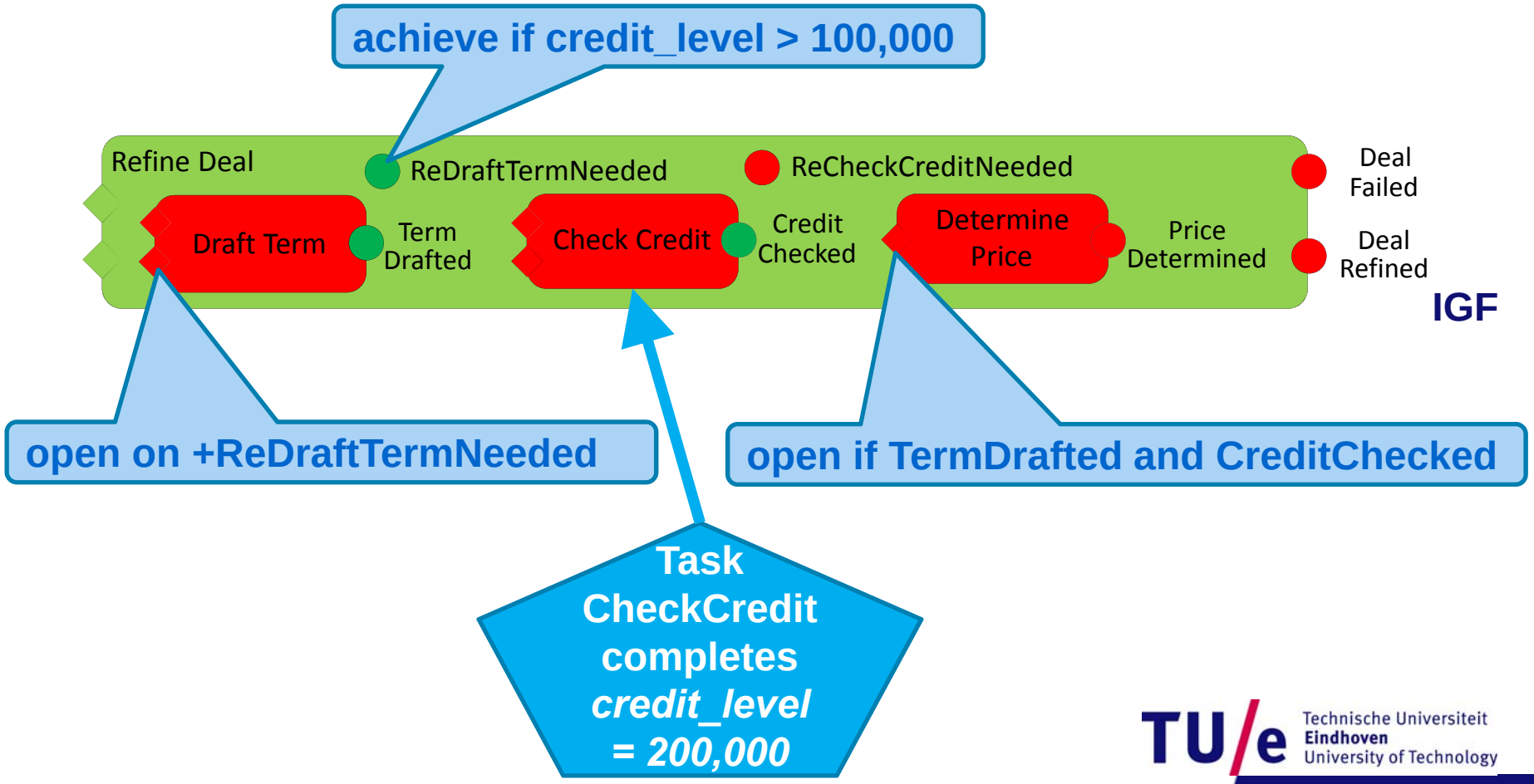
Continuing B-step (2)

- status is on
- status is off



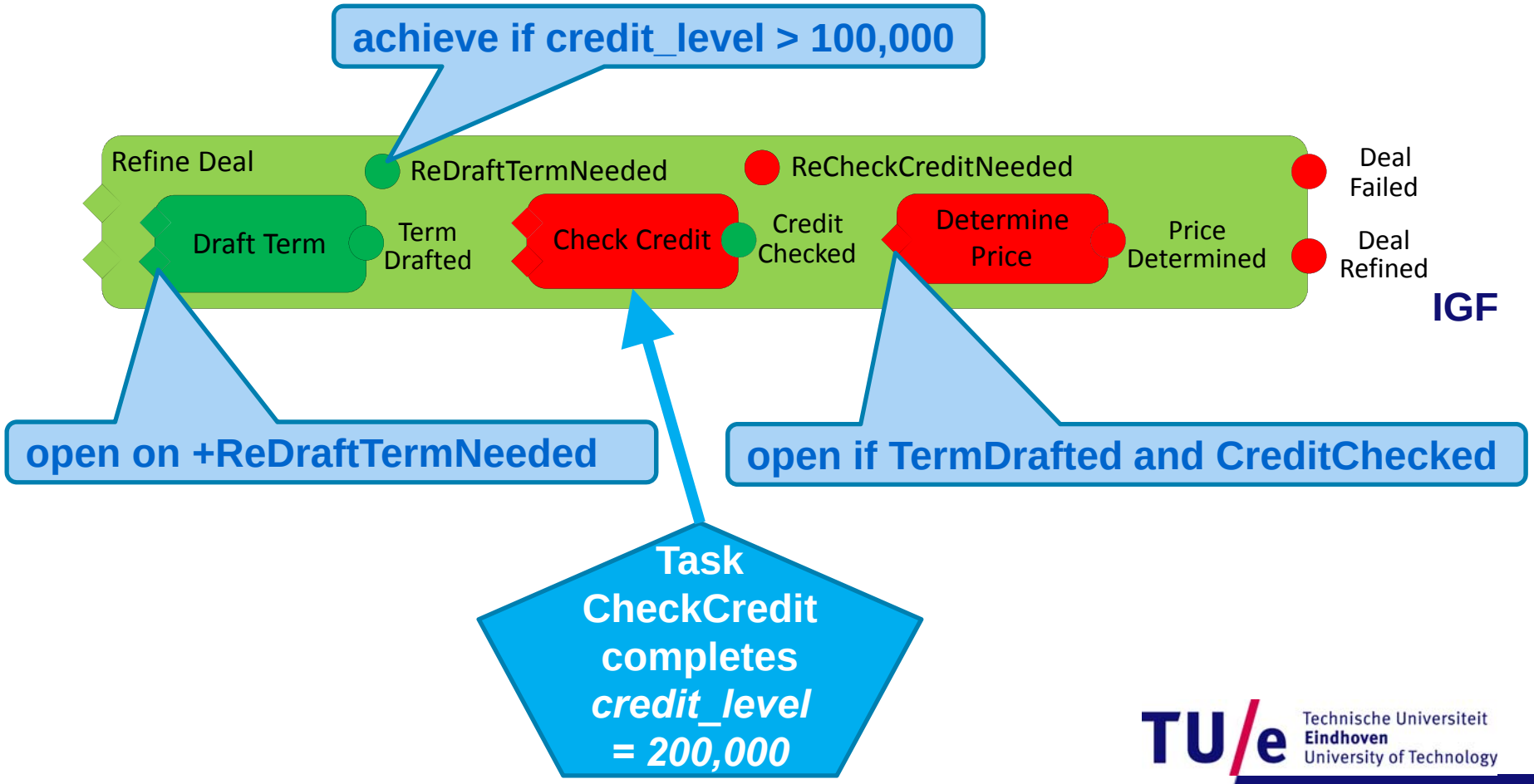
Continuing B-step (3)

- status is on
- status is off



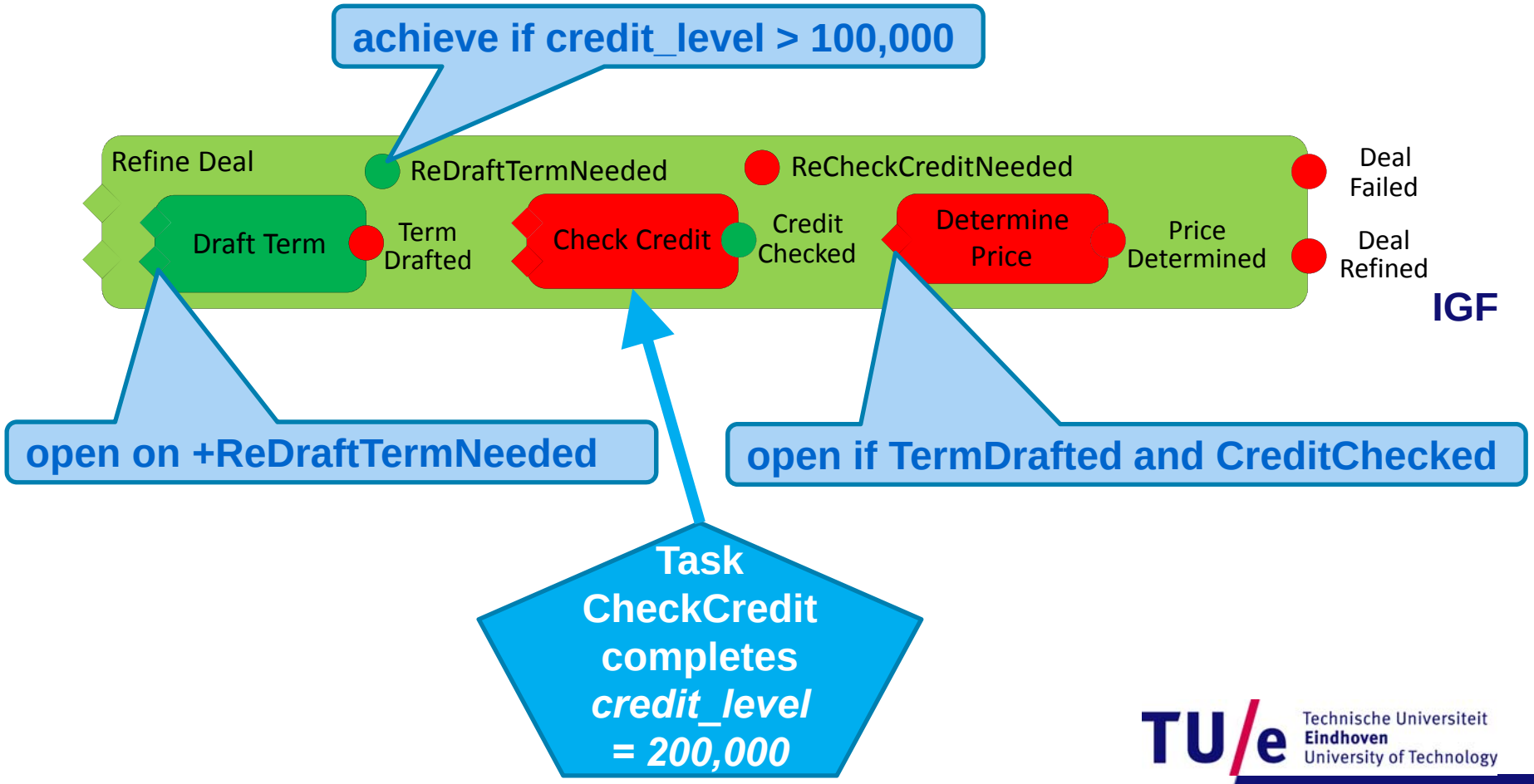
Continuing B-step (4)

- status is on
- status is off

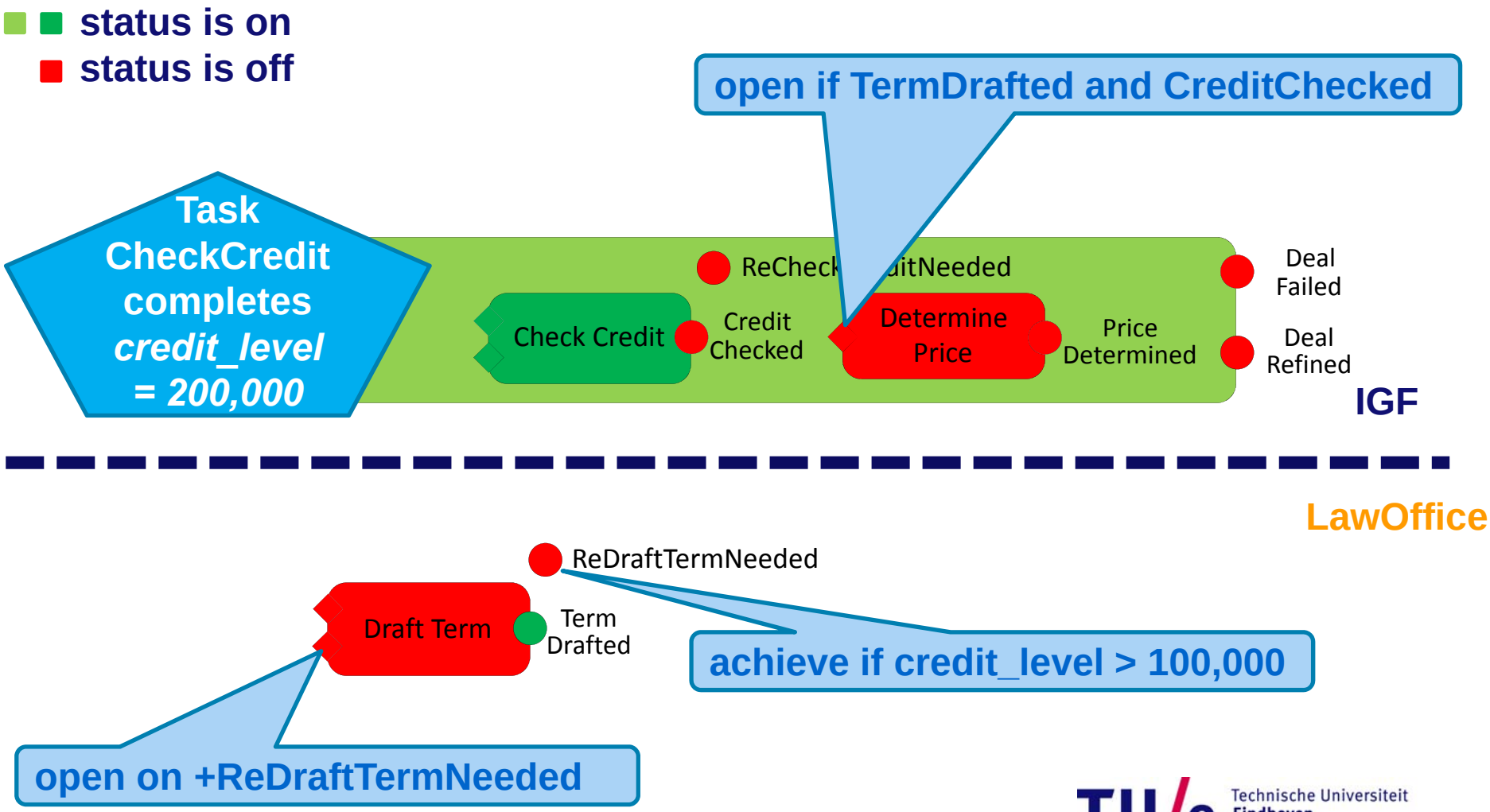


End of B-step (5)

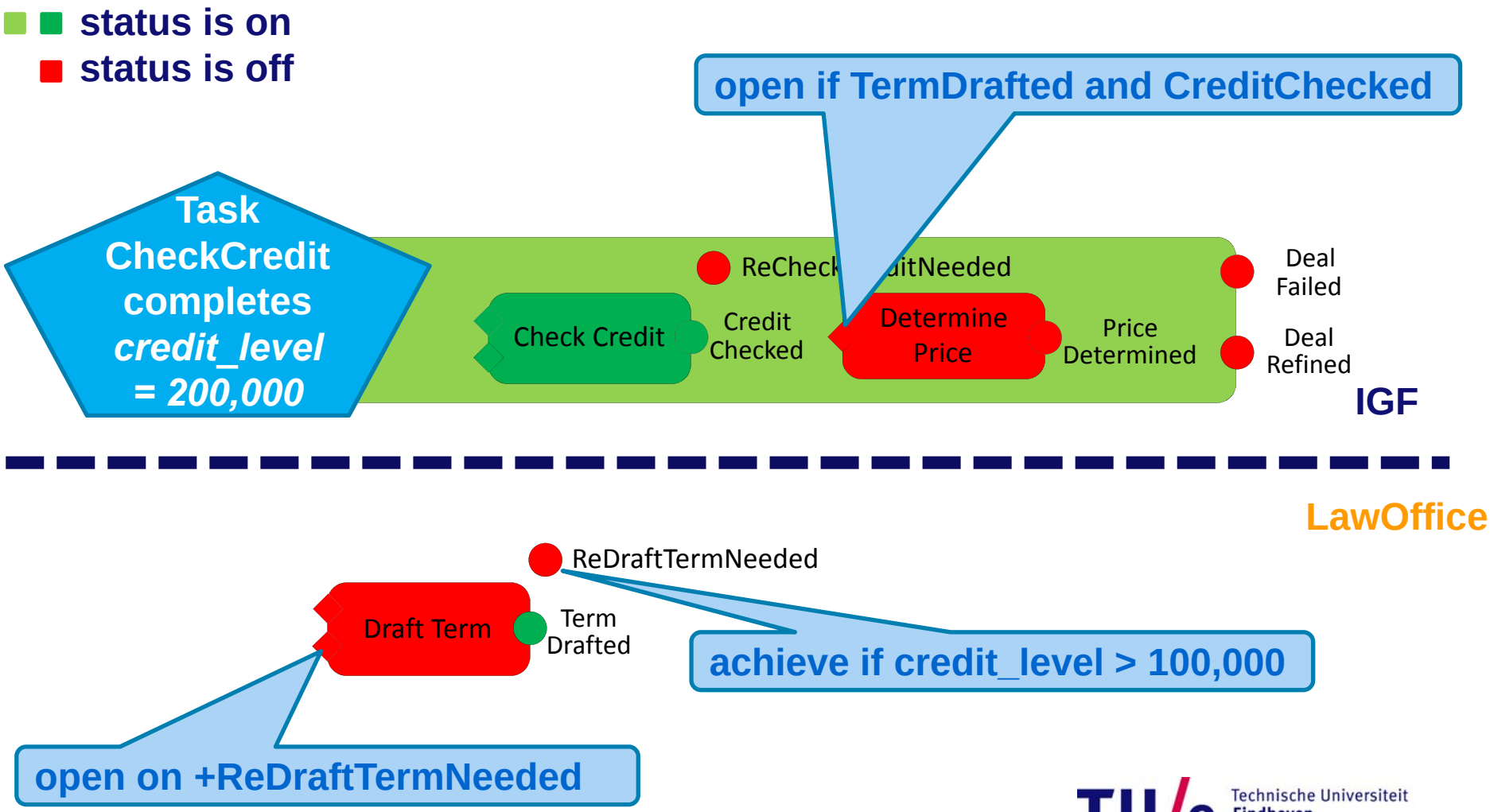
- status is on
- status is off



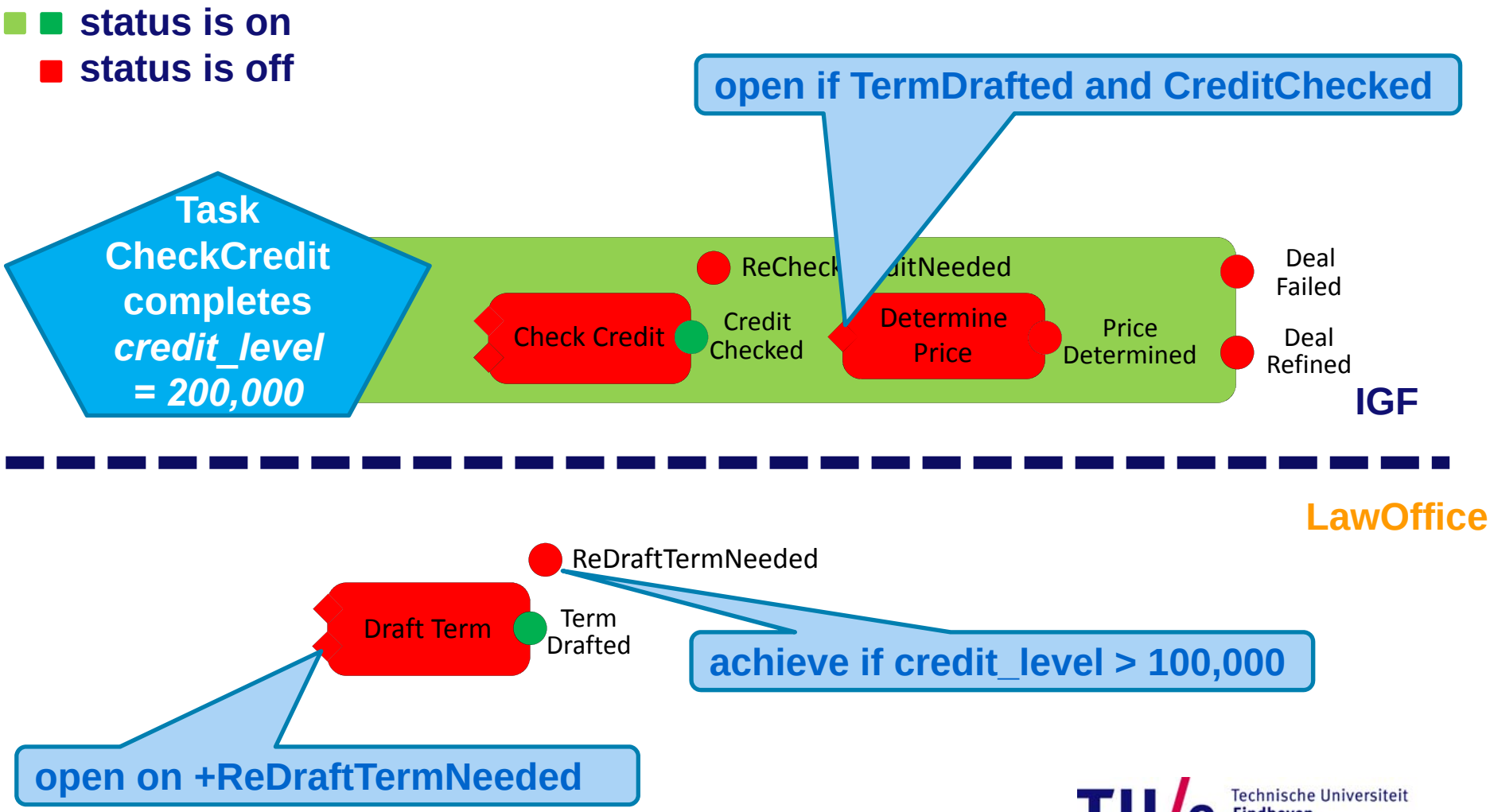
Splitting example scenario results in race condition



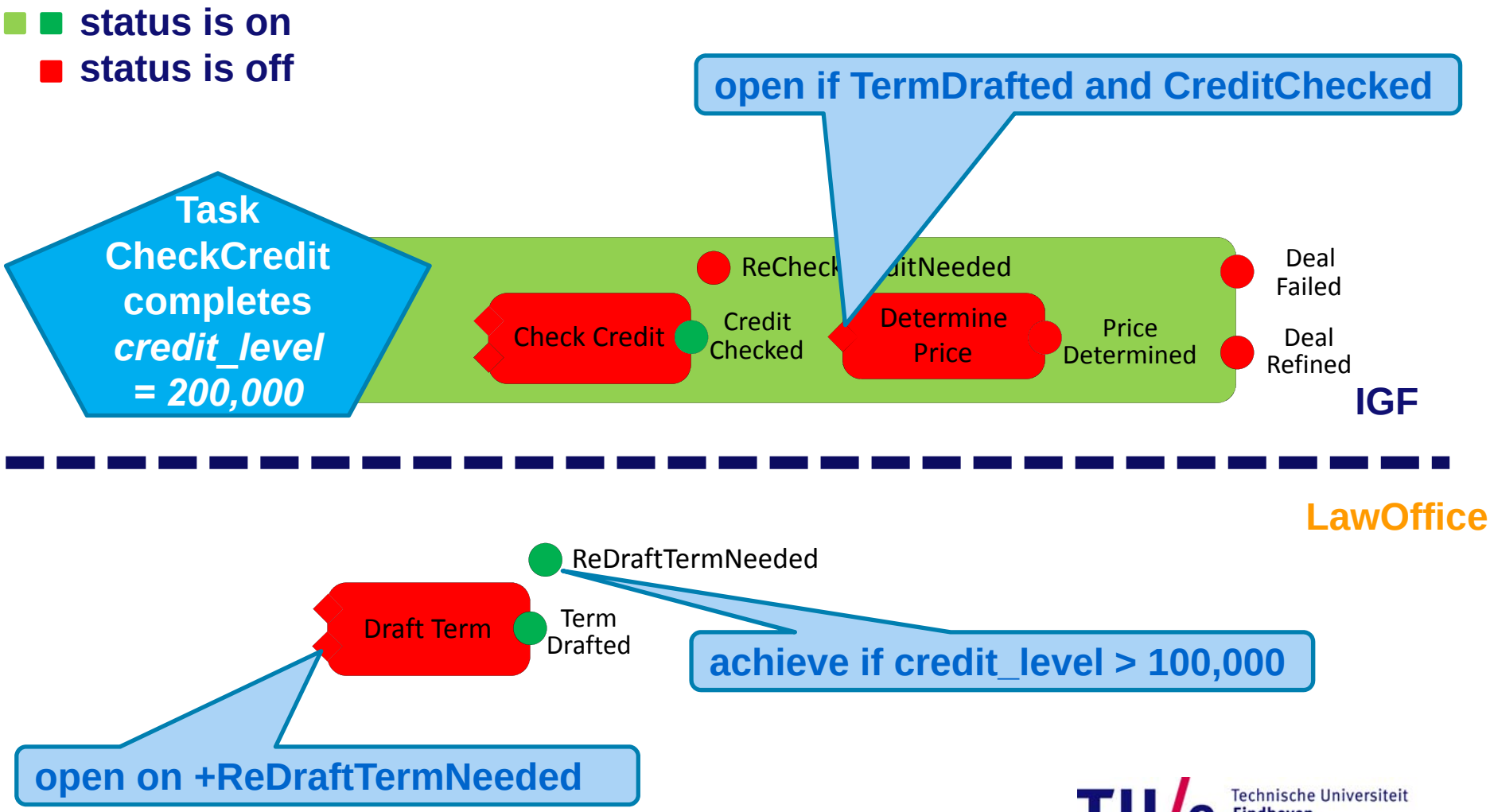
Splitting example scenario results in race condition



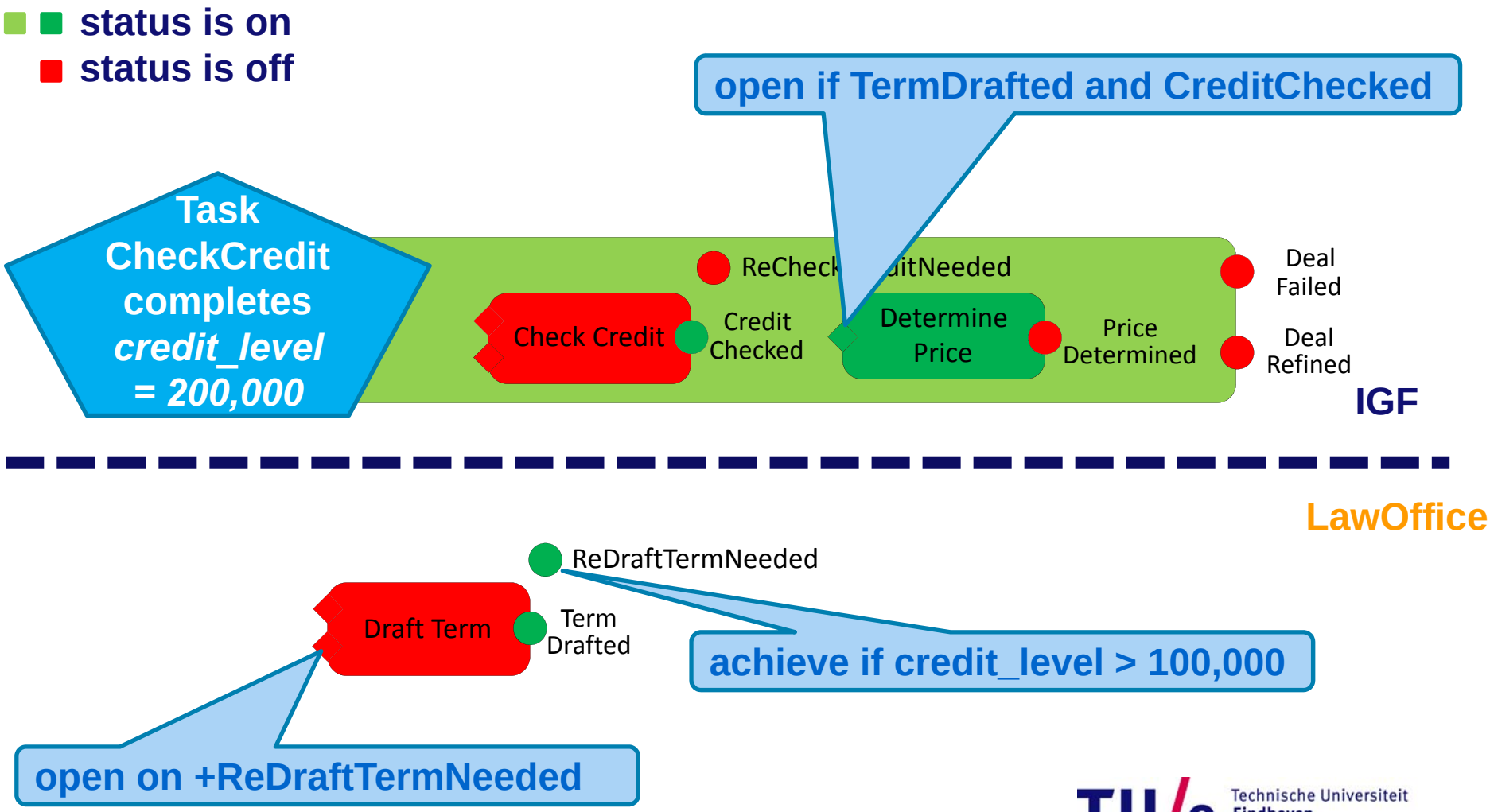
Splitting example scenario results in race condition



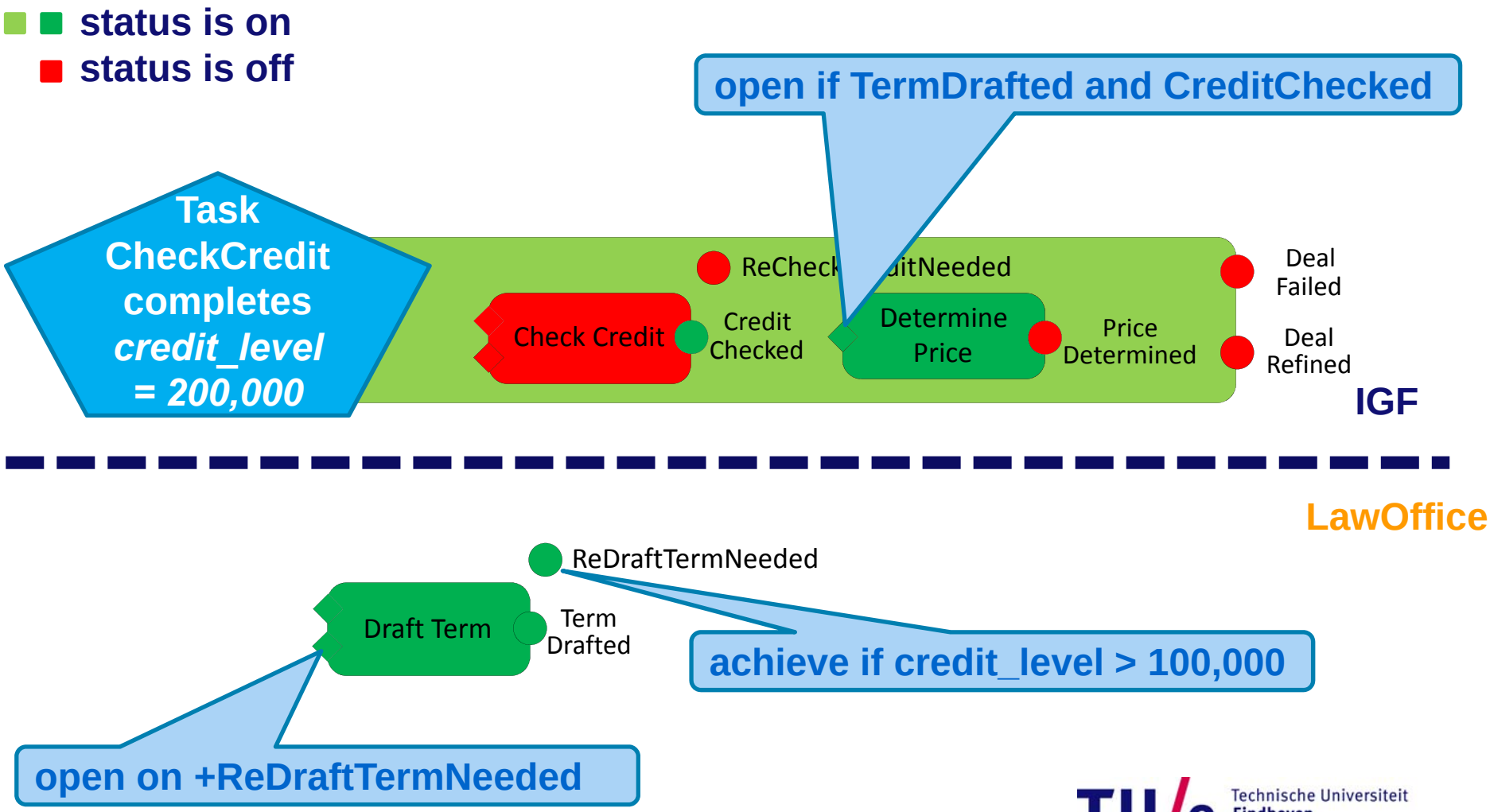
Splitting example scenario results in race condition



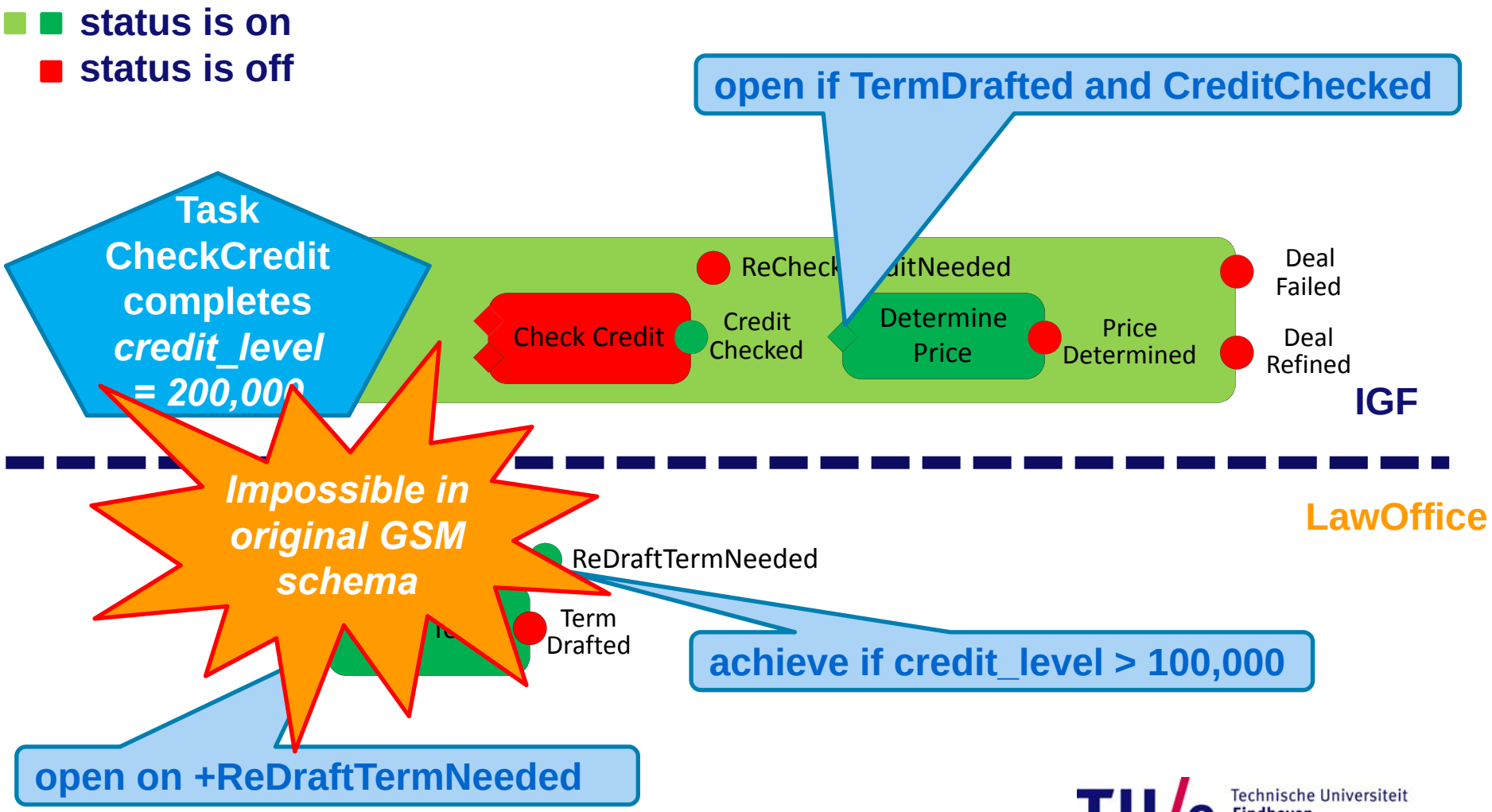
Splitting example scenario results in race condition



Splitting example scenario results in race condition

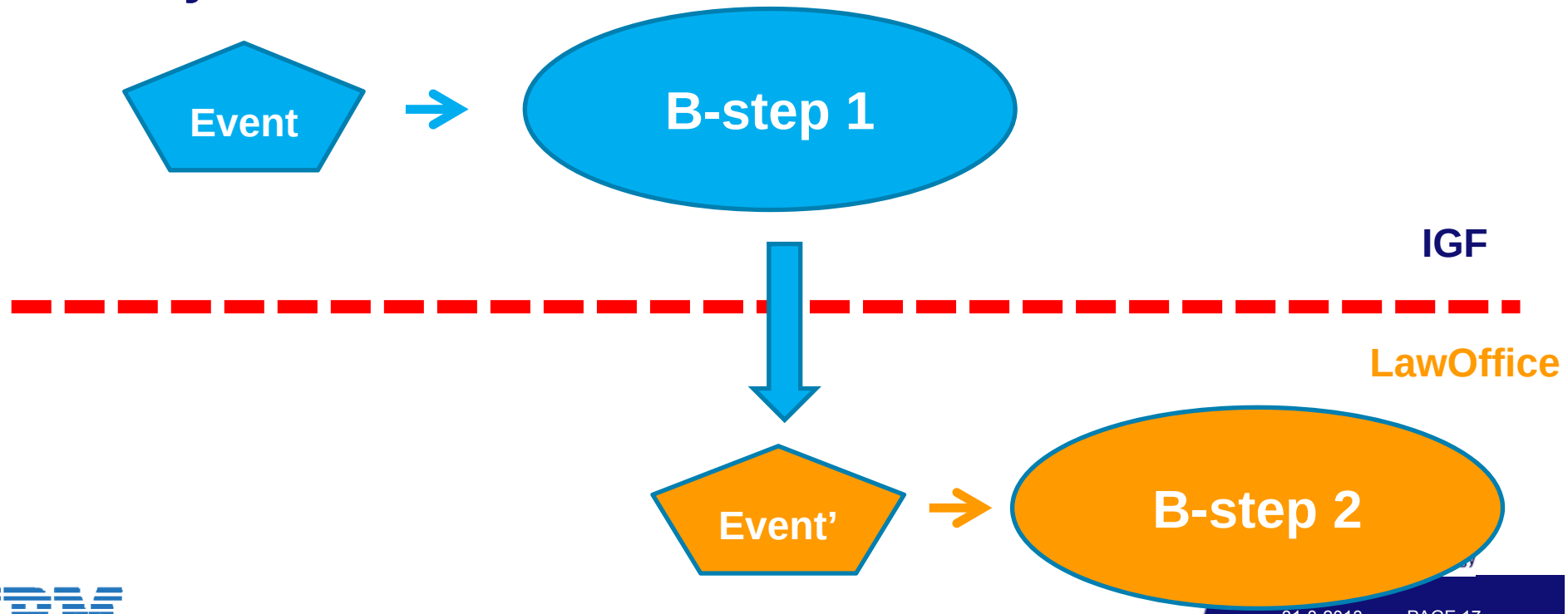


Splitting example scenario results in race condition

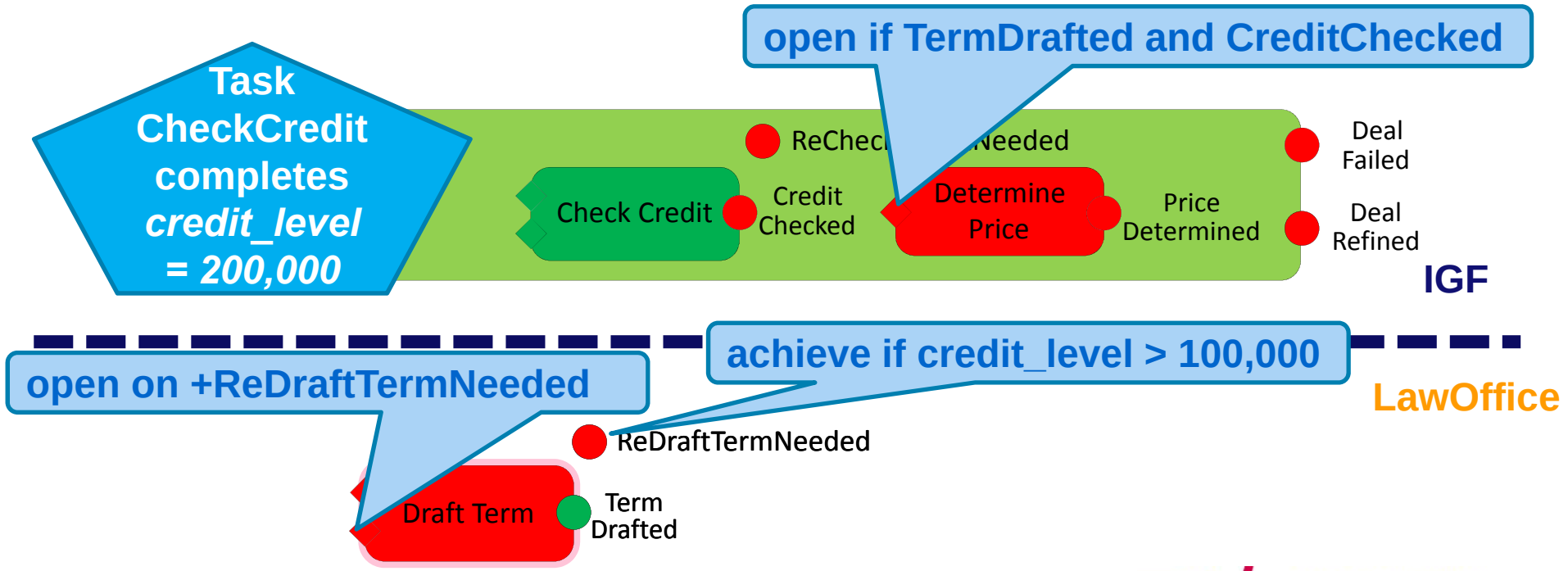


Possible solutions

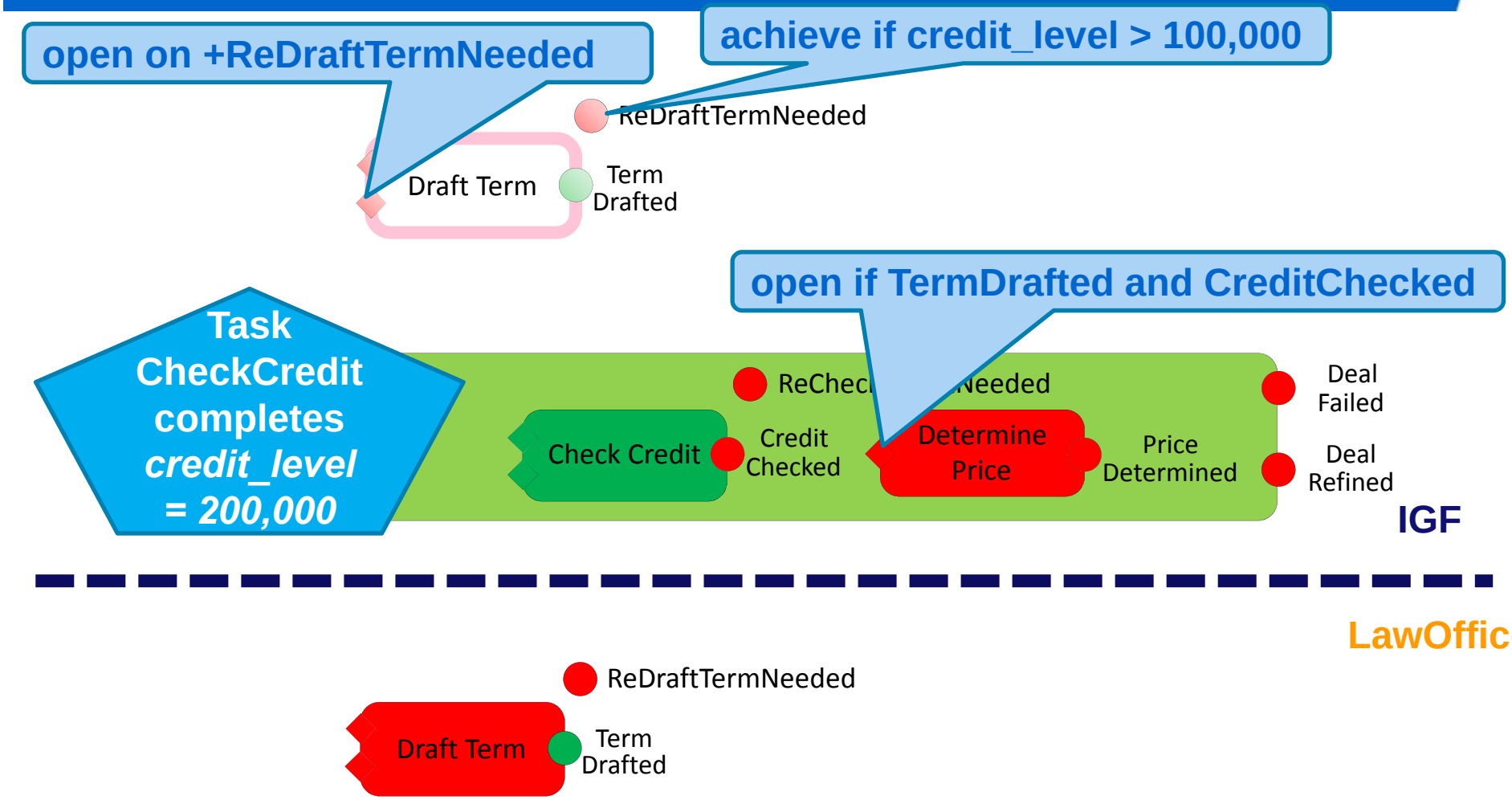
1. Extend local GSM engines with complex synchronization logic
2. Better: restructure GSM schema to avoid complex synchronization



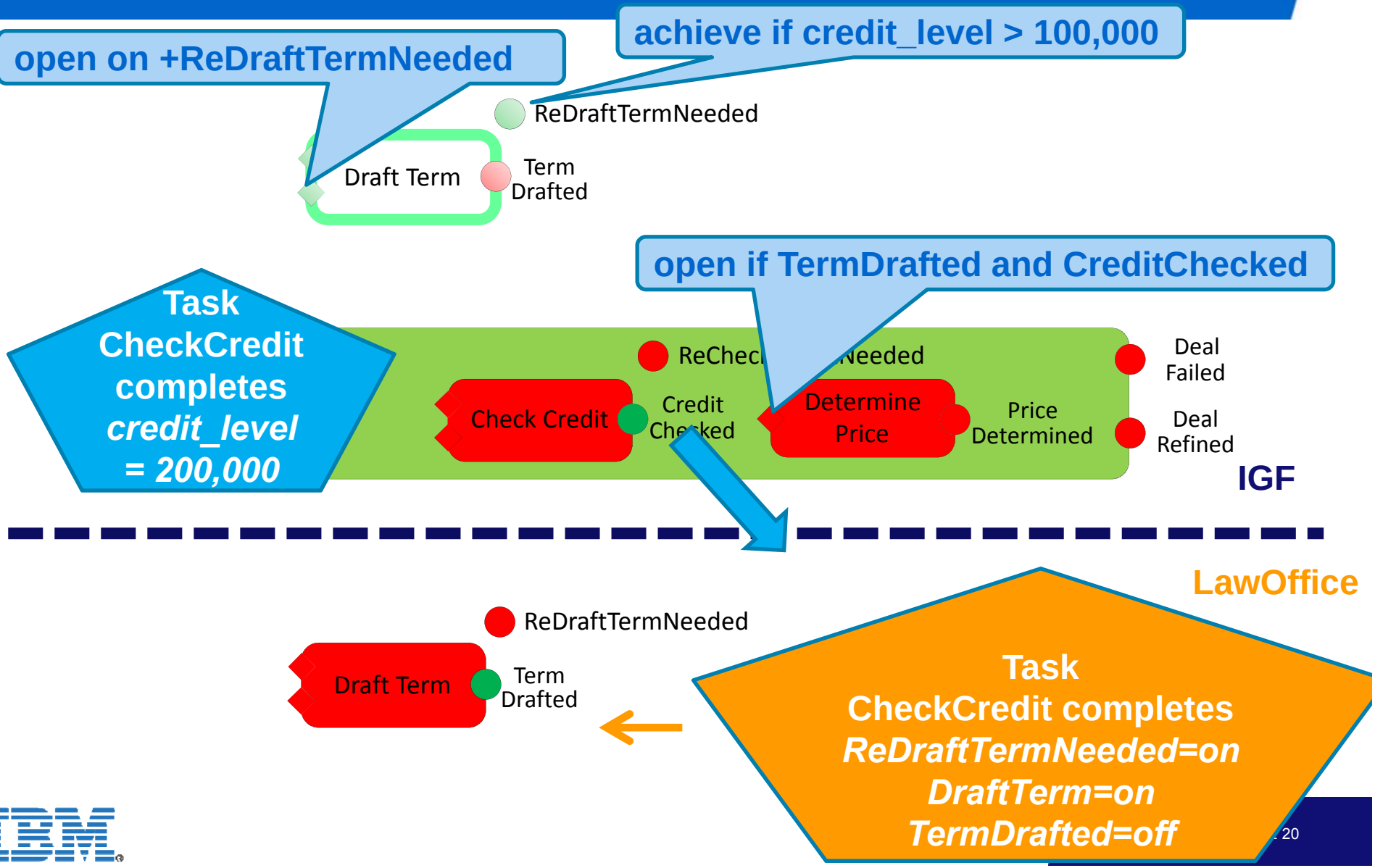
Resolved race condition



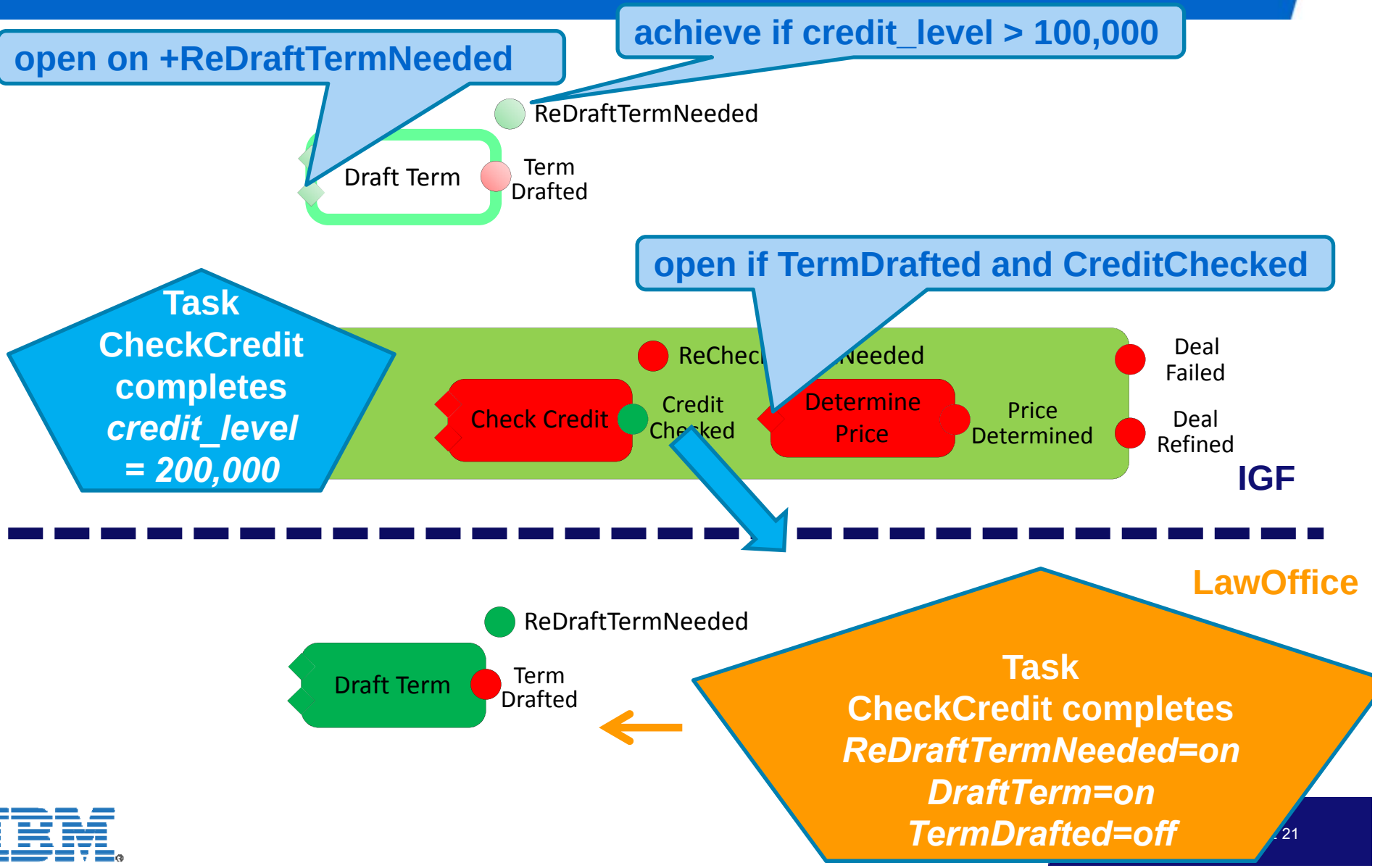
Resolved race condition



Resolved race condition: after B-step 1



Resolved race condition: after B-step 2



Solution approach

- Keep local cache of status of needed remote stages/milestone
  - determined per external event type
- Reallocate business rules to avoid complex synchronization
  - Sometimes: rule executes at different location than the corresponding stage/milestone
- Runtime simulation protocol based on distributed 2-phase commit
- Result (see paper): split GSM schema faithfully simulates original GSM schema

Formal results

Theorem 1

- Let Σ be a snaphot of GSM schema Γ ;
- Let schemas Γ_1 and Γ_2 be a “split” of schema Γ ;
- Let snapshots Σ_1 and Σ_2 be the split of Σ;
- Let e be an external event of type E;
- Then: the application of e on (Σ_1,Σ_2) faithfully simulates the application of e on Σ

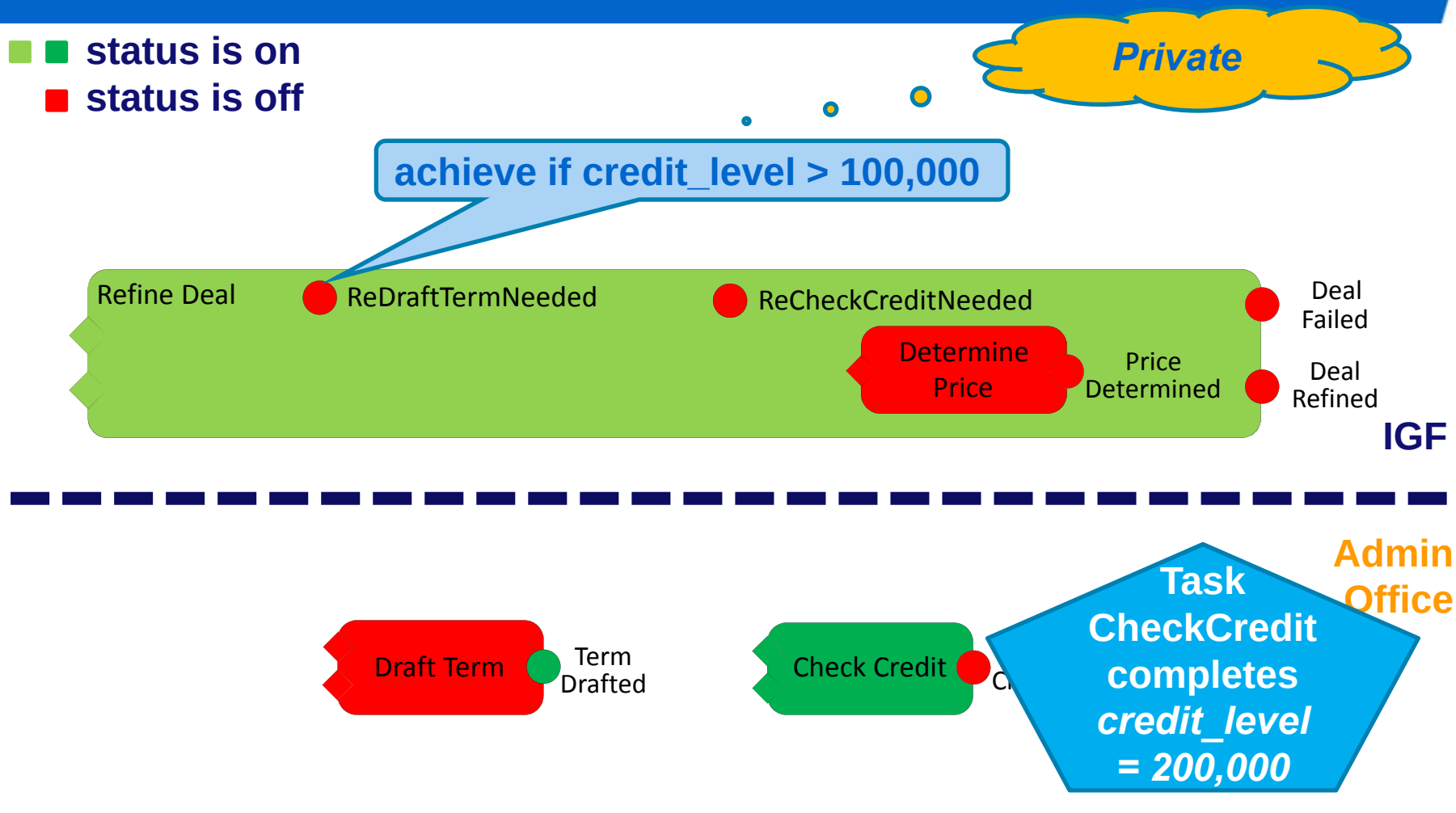
Theorem 2

- Using the “runtime simulation protocol” (see paper) the above result generalizes to streams of external events.

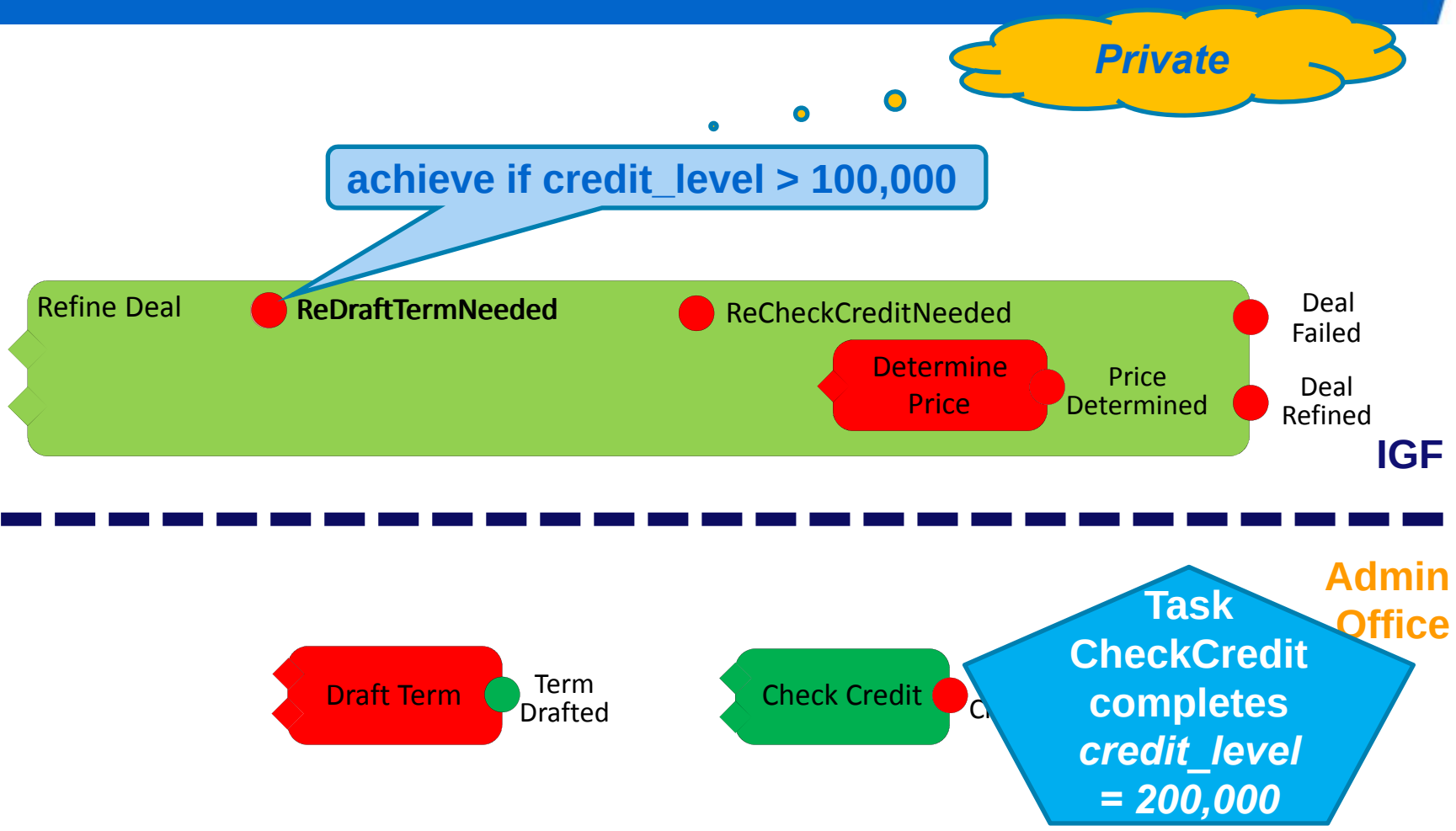
Yet another problem…

... revealing of secrets

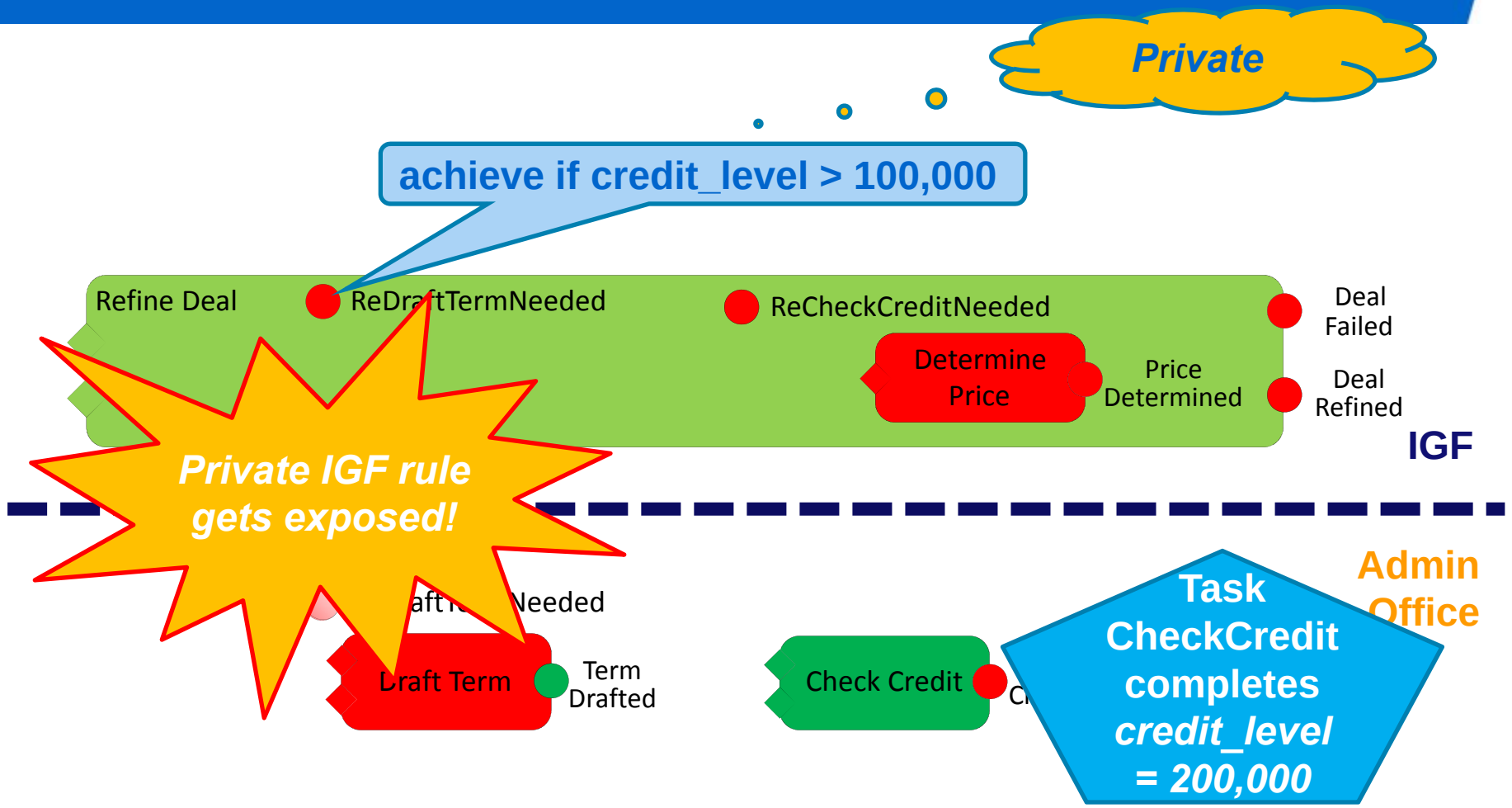
Alternative outsourcing scenario



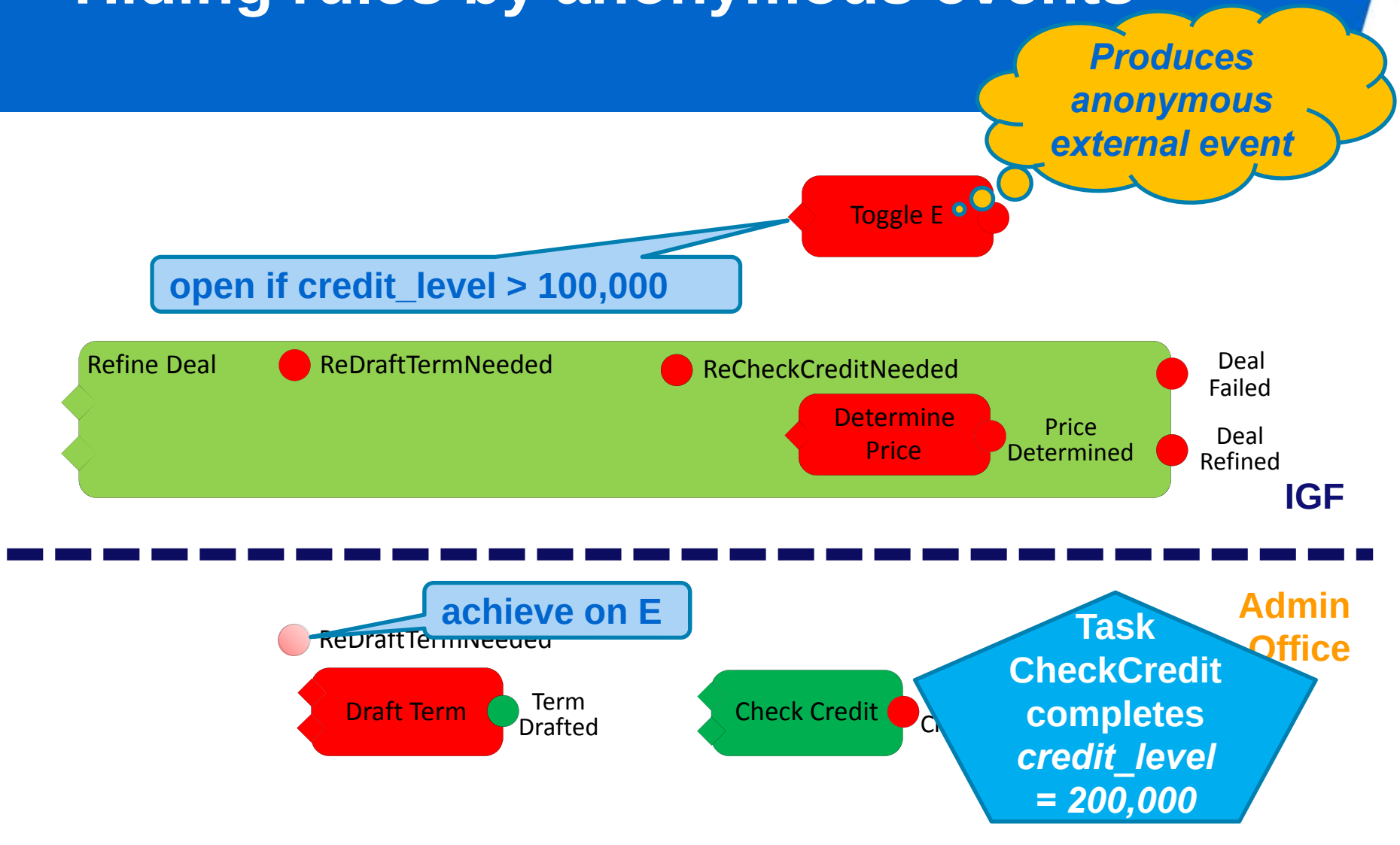
Restructured GSM schema



Restructured GSM schema



Hiding rules by anonymous events



Result on privacy

- Let *Γ* be GSM schema and let *H* be a set of attributes whose rules should be hidden.
- Construct *Γ' = hide(Γ,H)* as follows
  - For hiding the rules of *n* status attributes, in centralized GSM model
    - For each hidden status attribute *a*, create an anonymous event E_a
    - Each rule that triggers *a* is transformed into a rule that triggers a stage that generates E_a
- Lemma: each B-step of *Γ* is faithfully simulated by a cluster of *n* + 1 B-steps of *Γ'.*
- To perform splitting with hidden status attributes, create *Γ' = hide(Γ,H)* and split *Γ'*.

Complete algorithm

- **Input:**
  - **GSM schema Γ**
  - **desired split of stages/milestones**
  - **status attributes H whose rules are to be hidden**
- **Step 1: introduce anonymous events in $\Gamma \rightarrow \Gamma'$**
- **Step 2: perform splitting on $\Gamma' \rightarrow (\Gamma_1 , \Gamma_2)$**

- **Main result (see paper):**
  - **(Γ_1 , Γ_2) faithfully simulates Γ**
  - **(Γ_1 , Γ_2) does hide rules for status attributes in H**

Conclusion

- Comprehensive framework for supporting outsourcing of GSM schemas
- Covers both design-time and run-time
- Future work:
  - multi-party outsourcing
  - interacting artifact types
  - extend results to OMG's Case Management Modeling Notation

Thank you!