

**ABSTRACTIONS FOR
PROGRAMMING SIP
BACK-TO-BACK
USER AGENTS**

INTRODUCING . . .

**StratoSIP:
SIP at a very
high level**

Pamela Zave

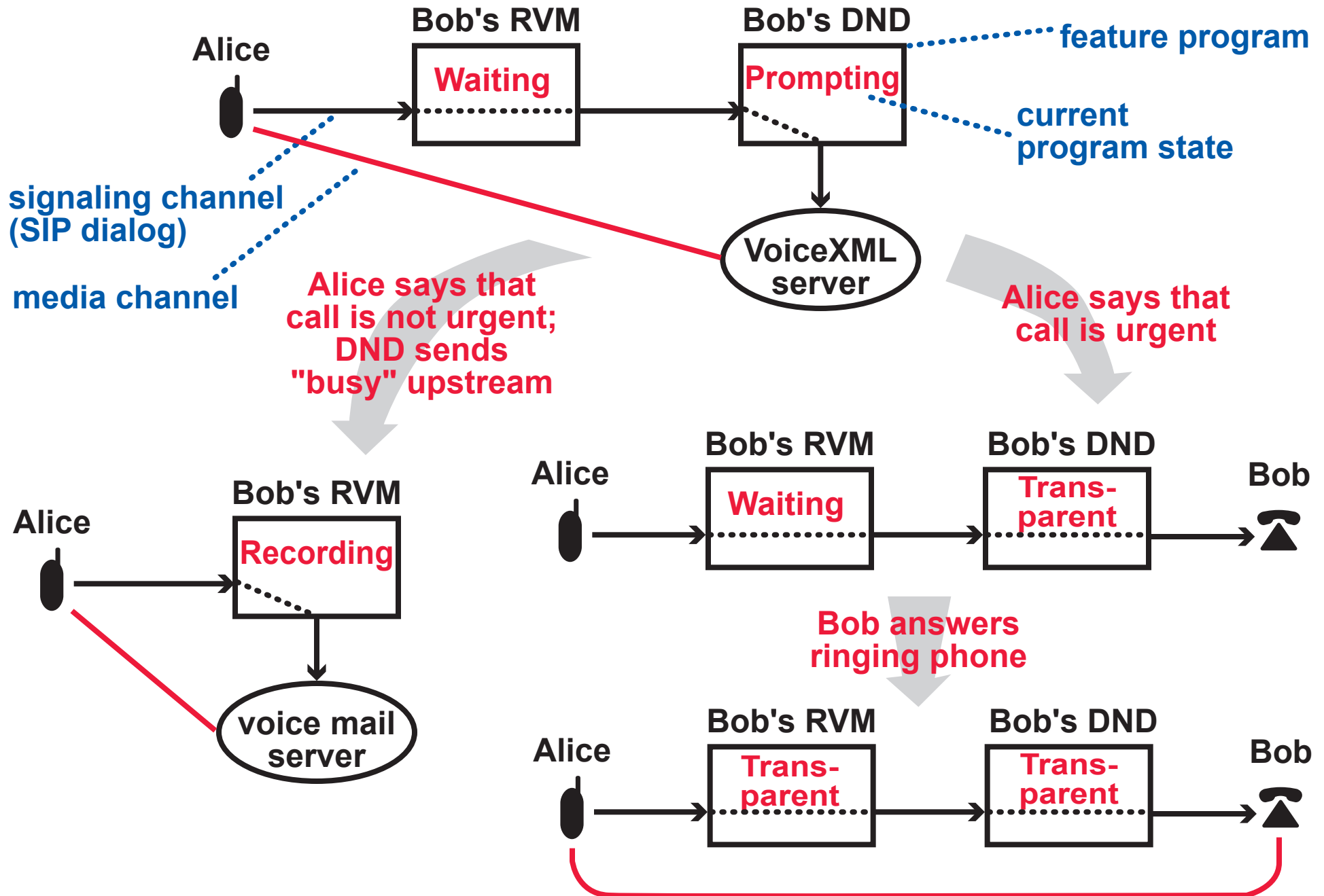
Gregory W. Bond

Eric Cheung

Thomas M. Smith

*AT&T Laboratories—Research
Florham Park, New Jersey*

RECORD VOICE MAIL AND DO NOT DISTURB AT RUNTIME



StratoSIP supports this programming style, which is . . .

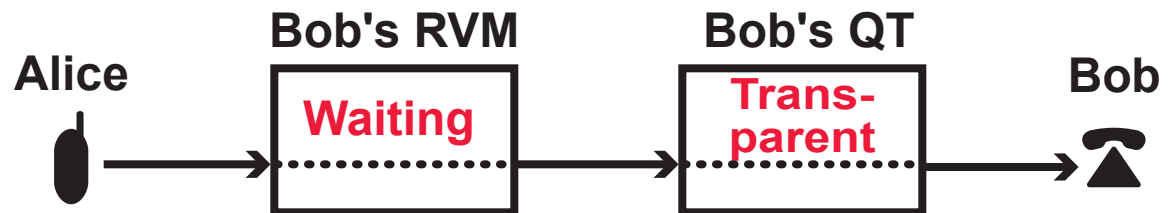
COMPOSITIONAL FEATURE PROGRAMMING

Each feature is an independent program.

Each feature works correctly by itself or composed with any combination of other features.

Feature interactions are predictable; they can be managed without changing the features themselves.

for example, by adjusting feature order, which is data in the feature container



at the top level, each feature program is a finite-state machine

a program has *autonomy*: when it has a function to perform, it performs it without external help

a program has *transparency*: when it has no function to perform, it is unobservable

a program is *context-independent*: it does not refer to other features

**THIS STYLE WAS INTRODUCED AND
PROVEN SUCCESSFUL BY THE
"DISTRIBUTED FEATURE COMPOSITION" (DFC)
ARCHITECTURE**

StratoSIP RUNTIME ENVIRONMENT

PROGRAMS ARE BACK-TO-BACK USER AGENTS . . .

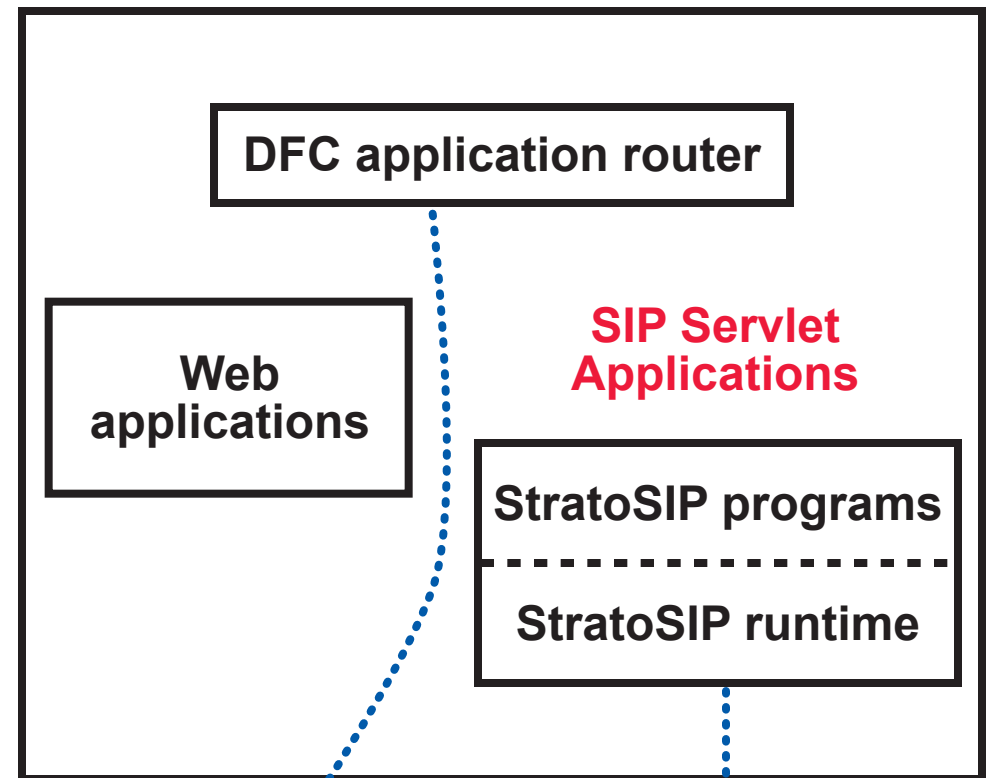
. . . which means that they are the endpoints of SIP dialogs.

However, contrary to some popular opinions about B2BUAs . . .

- they can run in user endpoints as well as application servers in the network
- they are easy to program (because of StratoSIP)
- they preserve the end-to-end behavior of SIP signaling, except when the program's purpose is to modify it

PROGRAMS ARE ALSO SIP SERVLETS

JSR 289 SIP Servlet Container

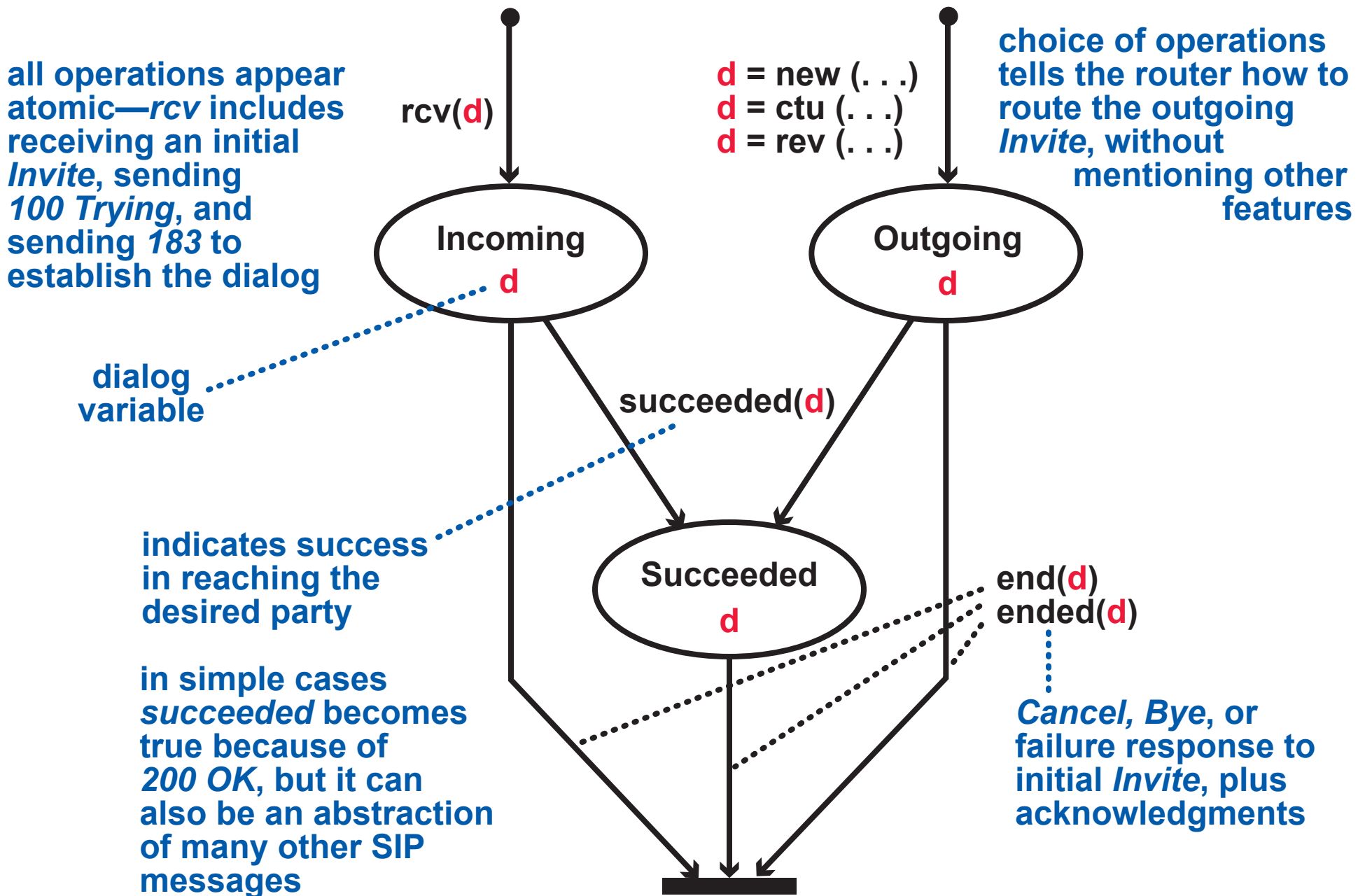


all components of
the runtime environment
are available as open-source software

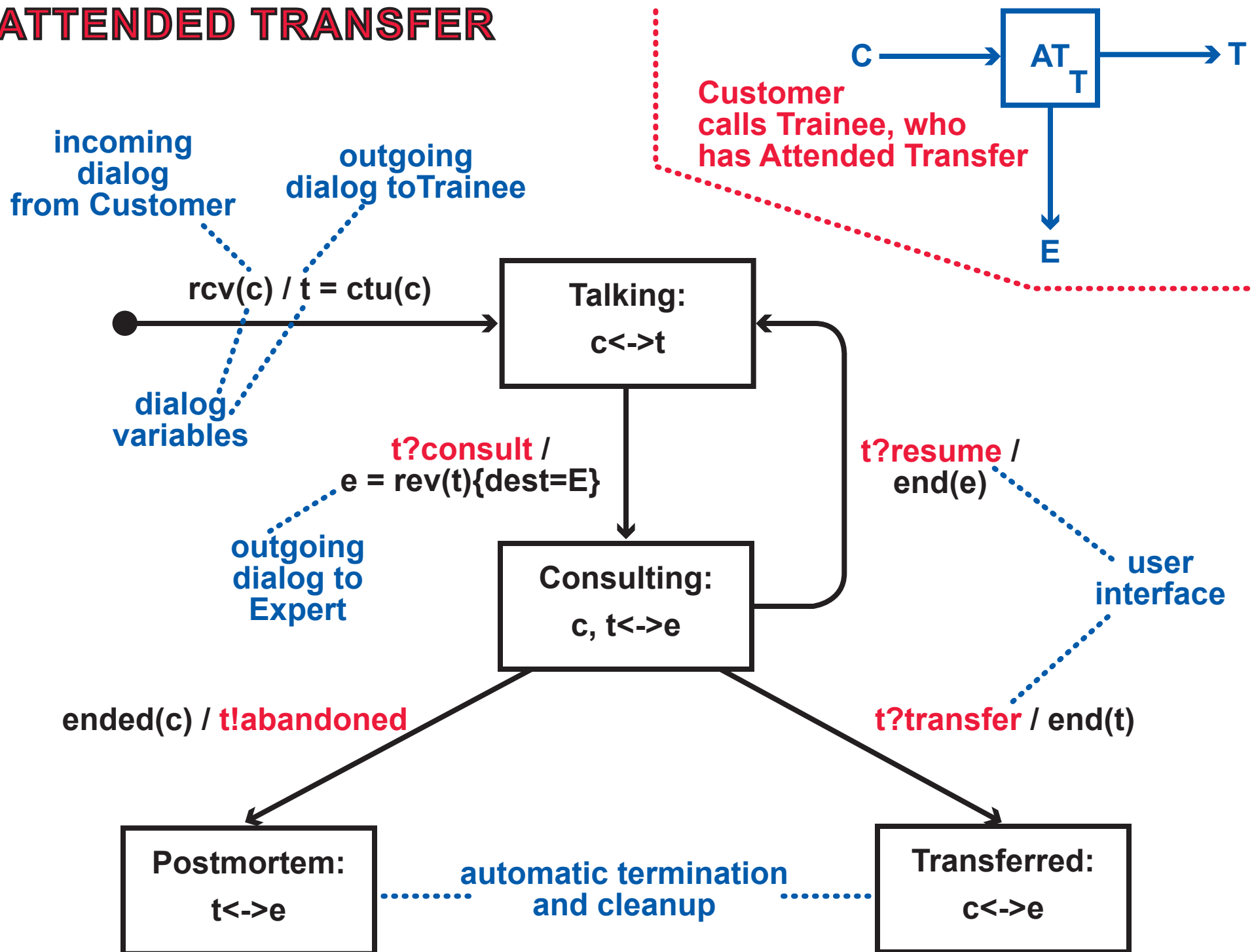
assembles graph
structures of programs
and signaling channels

implementation
in ECharts and
Java

StratoSIP DIALOG ABSTRACTION



ATTENDED TRANSFER



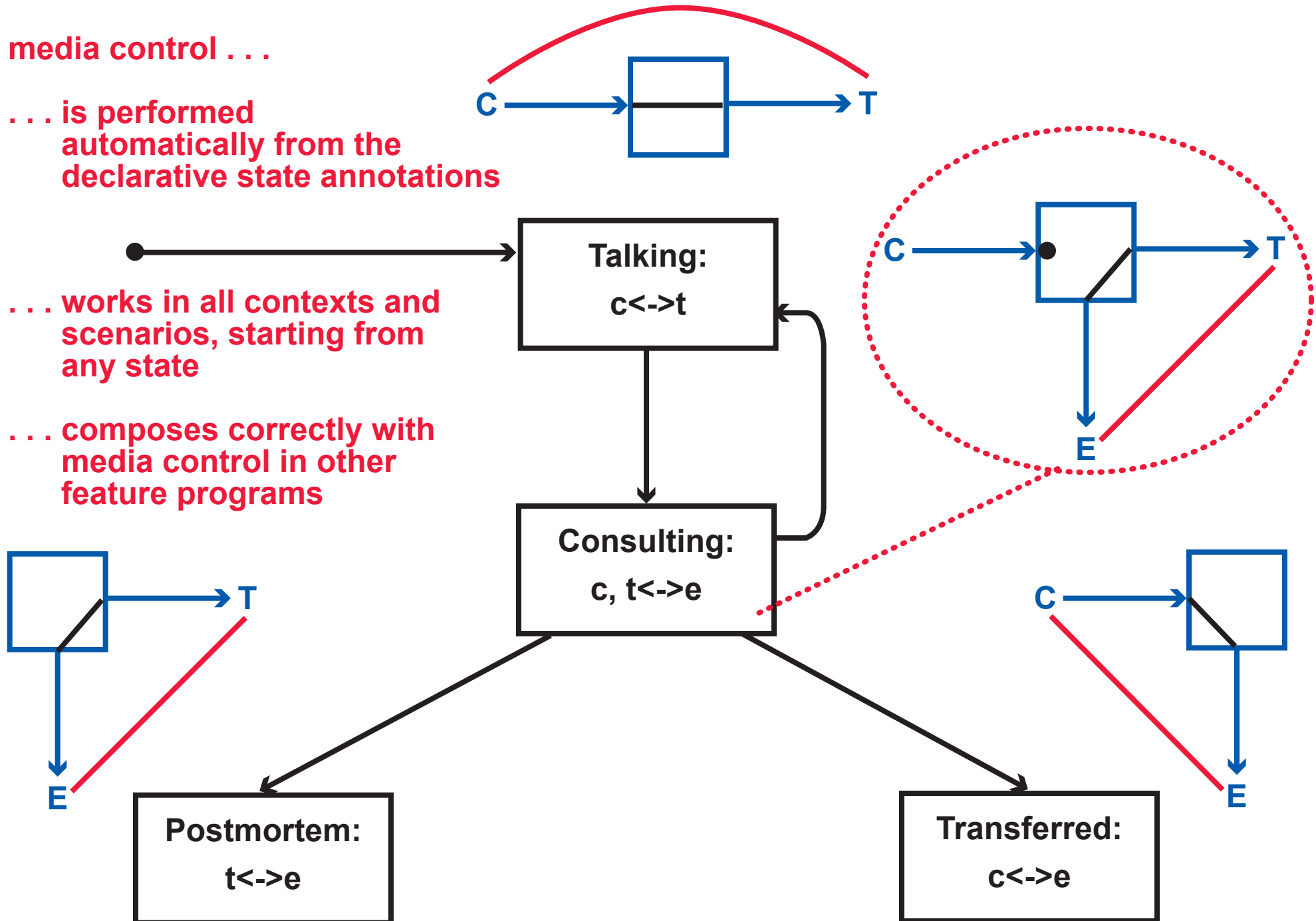
MEDIA CONTROL

media control ...

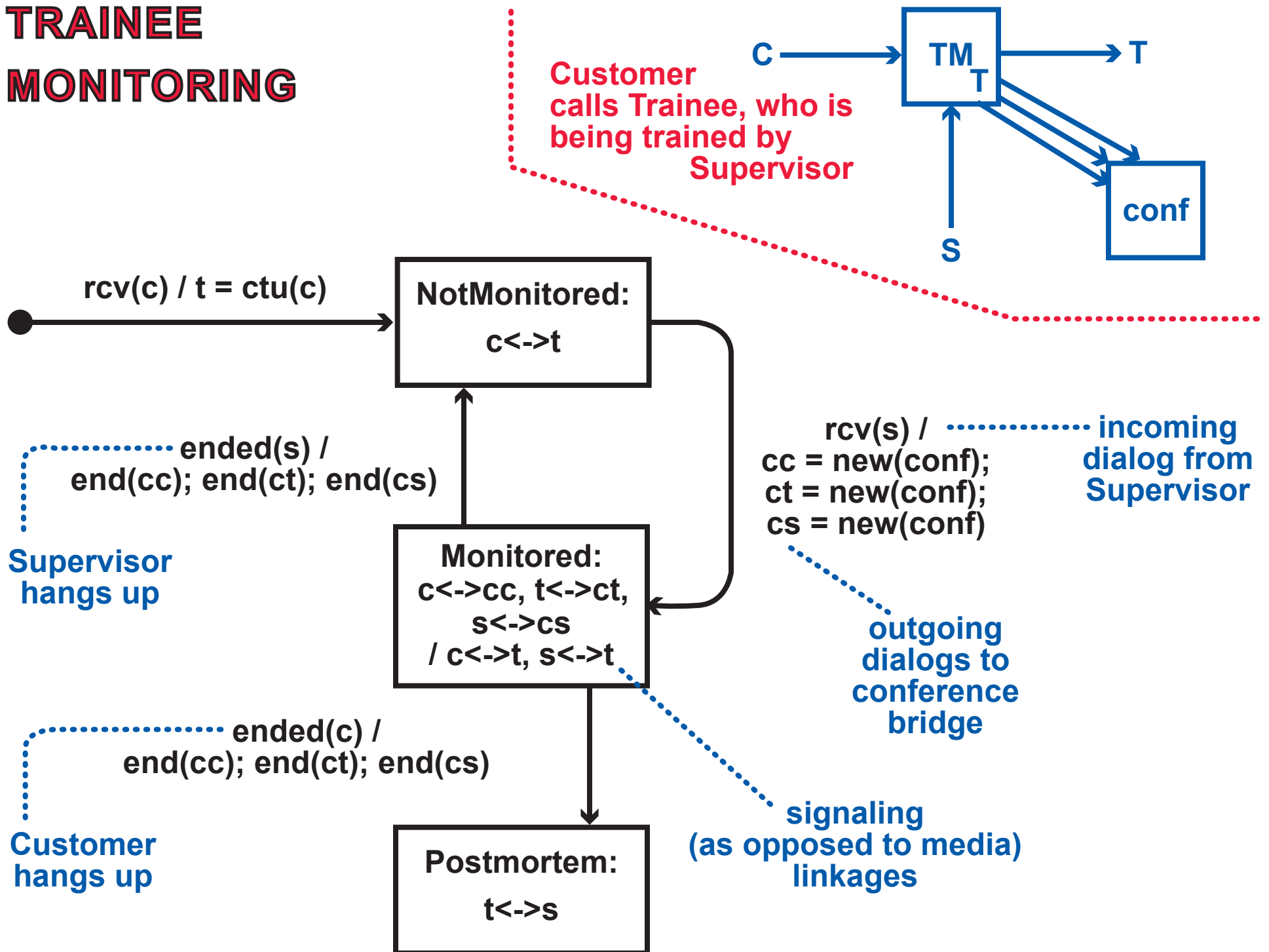
... is performed automatically from the declarative state annotations

... works in all contexts and scenarios, starting from any state

... composes correctly with media control in other feature programs

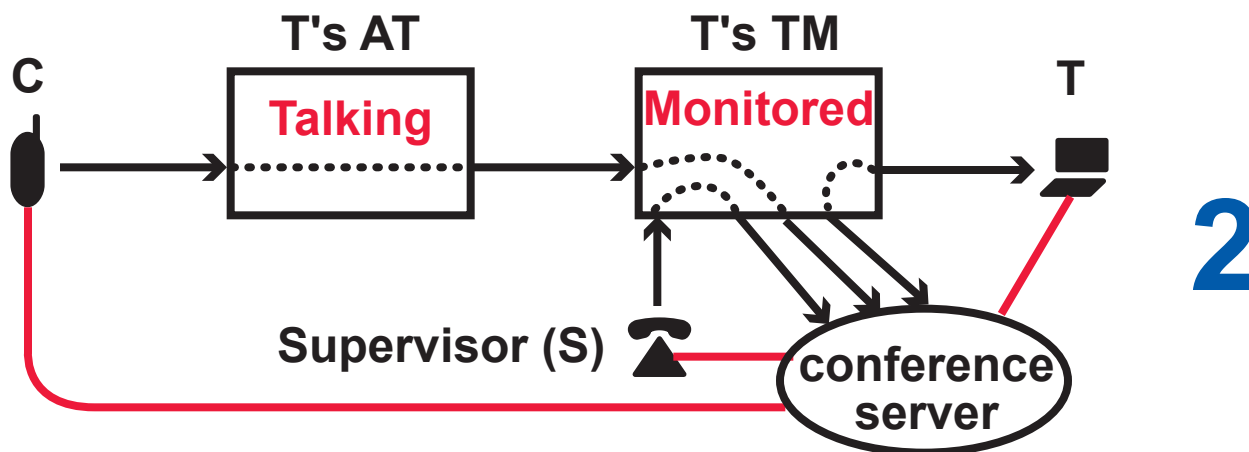
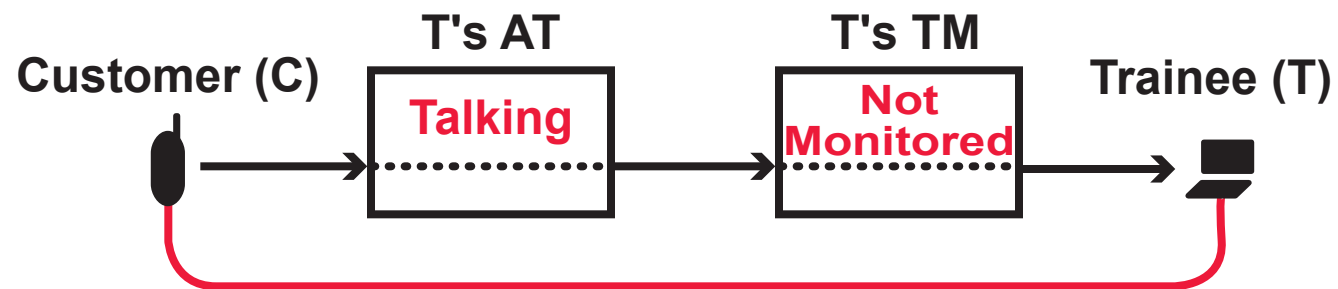


TRAINEE MONITORING



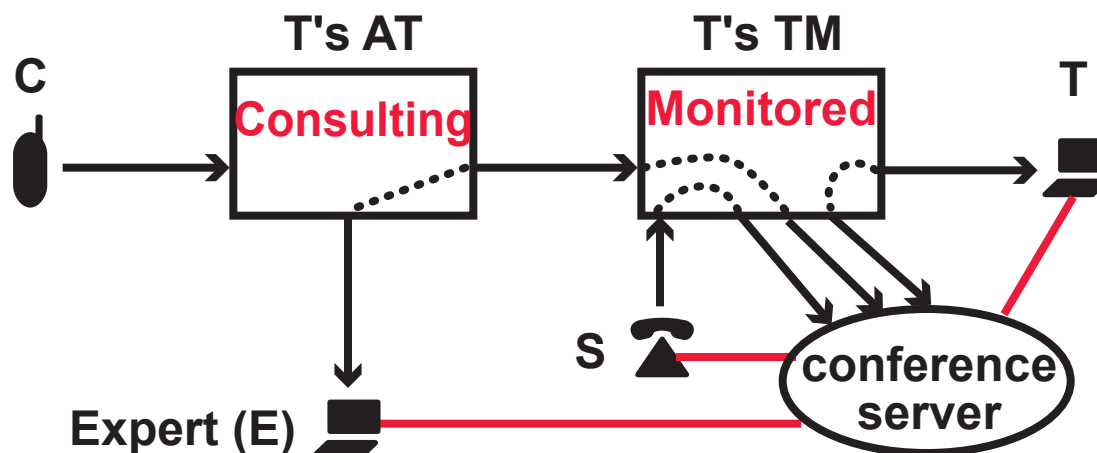
ATTENDED TRANSFER AND TRAINEE MONITORING

1

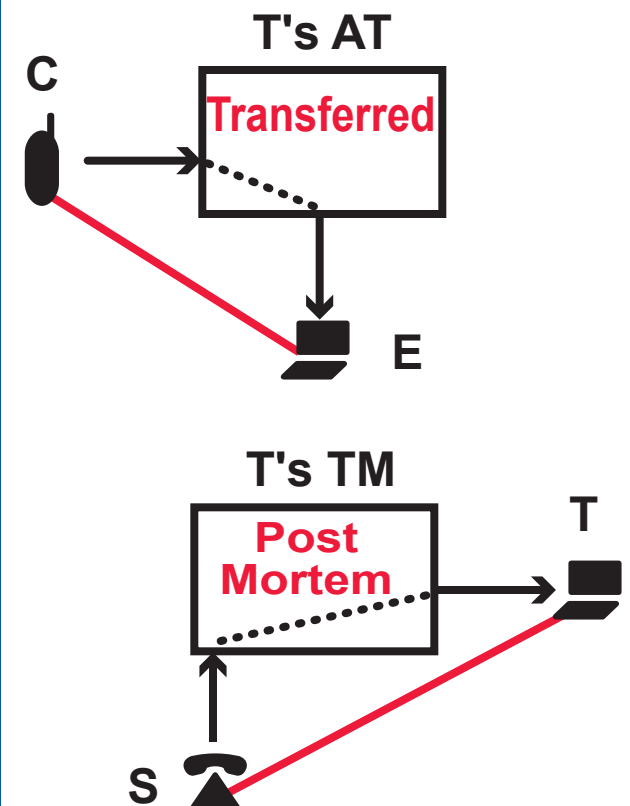


2

3



4



COMPOSITIONAL MEDIA CONTROL

**THIS IS THE CORE TECHNOLOGY OF
StratoSIP**

**WE DO NOT YET KNOW WHETHER
IT IS COMPATIBLE WITH MEDIA
CONTROL USING Refer MESSAGES,
Replaces AND Join HEADERS**

- implemented using the "third-party call control" style of SIP
- completely general with respect to feature functions
- compositional semantics has been specified formally in temporal logic
- implementation has been verified using model-checking
- based on 9 years of research
- turns a very difficult kind of programming into something safe and easy, accessible to (for example) Web programmers

COMPARISON

	Compositional Media Control	Refer/Replaces/Join
What version of SIP must endpoints implement?	basic SIP (RFC 3261)	RFC 3261 plus 3 extension RFCs
Can feature programs be located in either endpoints or servers?	YES	YES
When independently-developed features are used together, what must be done so that the composed features control media correctly?	nothing	rewrite each feature program with extra cases to coordinate with the actions of the other
How many end-to-end signaling messages are required for the sample scenario?	48	81
Are there any other inefficiencies?	in some scenarios there are signaling hairpins (not media hairpins!) we think this can be fixed	NO

ONGOING WORK

IMPLEMENTATION

- finishing implementation
- testing with JSR 289 SIP Servlet containers (OCCAS, SailFin, Mobicents)
- hope to release as open-source code

SUPPORT FOR CONVERGED SERVICES

- we now have automatically-generated Web interfaces to feature and routing data
- automatically generate Web interfaces for feature events (commands and status reports)
- explore deeper levels of convergence

FUTURE WORK

LANGUAGE EXTENSIONS

- support for an arbitrary number of dialogs
- separate control of multimedia channels

SIP INTEGRATION

- handle a few additional requests introduced in SIP extensions (straightforward)
- look for safe ways to handle *Refer/Replaces/Join*