EXPERIENCE WITH

COMPONENT-BASED DEVELOPMENT

OF A TELECOMMUNICATION SERVICE

from AT&T Laboratories—Research:

Gregory W. Bond

Eric Cheung

Healfdene H. Goguen

Karrie J. Hanson

Don Henderson

Gerald M. Karam

K. Hal Purdy

Thomas M. Smith

Pamela Zave

1 This is a significant example of CBSE.

2 This is not object-oriented programming.

3 The component architecture works really well.

4 The component architecture is starting to become influential.

AT&T CALLVANTAGE[SM] SERVICE

IS A SIGNIFICANT EXAMPLE OF CBSE

- It is a consumer, broadband, voice-over-Internet-Protocol (VoIP) service.
- We built, deployed, and extended the advanced features using component-based technology.
- We support about 100,000 customers nationwide.
- The service has won two industry awards, citing its voice quality and advanced features.

WHAT DOES CALLVANTAGE DO?

Here are the commercials made by Oscar-winning documentary director Errol Morris, and starring three of the co-authors of this paper.

This is not object-oriented programming.

DISTRIBUTED FEATURE COMPOSITION (DFC)

. . . is an architecture for telecommunication services, designed with the goals of:

- feature modularity
- structured feature composition
- management of feature interactions

. . . is an adaptation of the *pipes-and-filters* architecture to the domain of telecommunications.

The architecture is implemented with object-oriented programming, so it is a higher-level, more domain-specific abstraction.

DISTRIBUTED FEATURE COMPOSITION (DFC)

a box/component is a concurrent process, implements a feature

connectors between components are "little telephone calls"

unbounded queues are necessary for signaling (rather than some more synchronous connector) because new stimuli can come from either end or the middle

MODULARITY IN A PIPES-AND-FILTERS ARCHITECTURE:

optionality: **every box is optional**

autonomy: **whenever a box has a function to perform, it can perform it without help**

transparency: **whenever a box has no function to perform, it is unobservable**

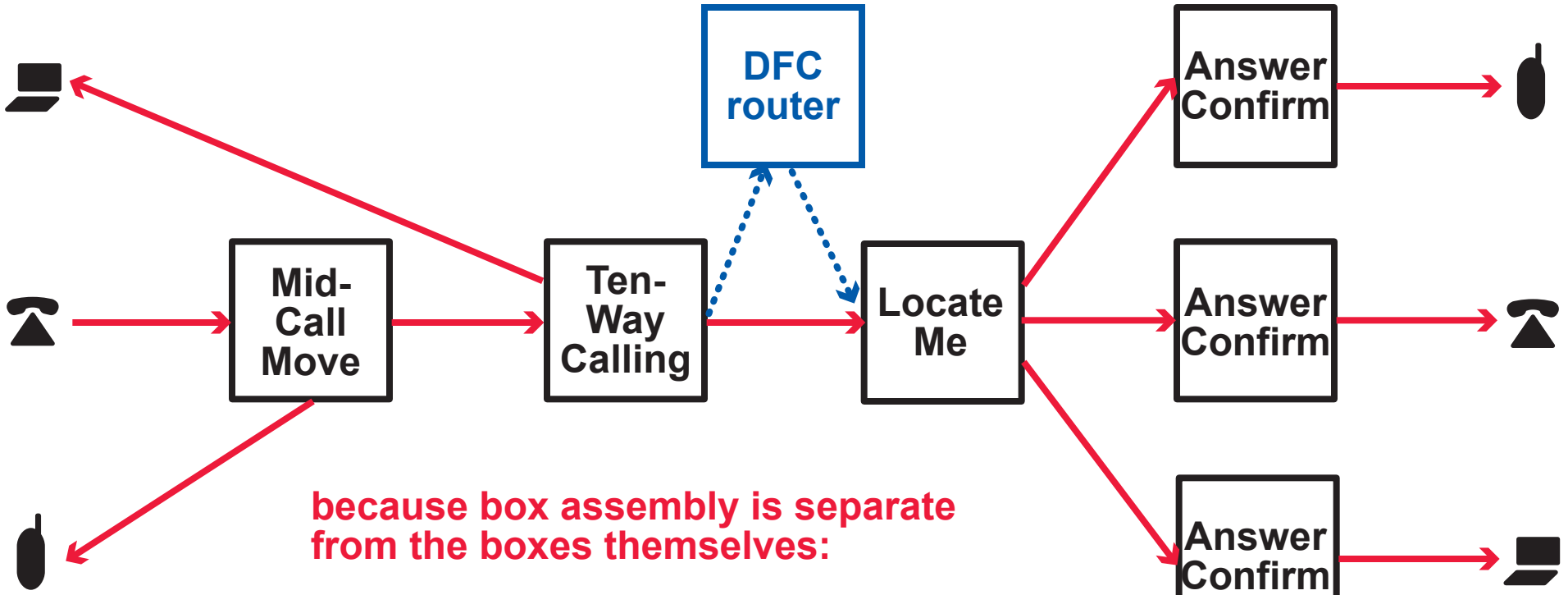
context-independence: **a box does not know or need to know what is at the other end of its calls**

DISTRIBUTED FEATURE COMPOSITION (DFC)

the graph of boxes and calls is assembled dynamically and evolves over time

a box can affect assembly by making highly constrained changes to setup signals

the assembly mechanism is the DFC router, which chooses a box based on source and target addresses, feature subscriptions, feature precedences, and history carried in the setup signal



because box assembly is separate from the boxes themselves:

- boxes know nothing about other boxes
- the assembly mechanism is powerful and well-suited to telecommunications

USE OF COMPONENTS

A typical call between two subscribers would have something like 20 feature boxes.

BOXES ARE IDENTIFIED FEATURES

15 identified and named features of the service are implemented by feature box classes.

The default situation is one feature box class per feature.

BOXES ARE REUSABLE BUILDING BLOCKS

There is reuse at several levels:

- **reuse of features from previous services**
- **reuse of generic box classes as components of complex features**
- **reuse of parameterized programs to make new box classes**

BOXES ARE ADAPTORS

The DFC-based application server operates in an environment with many other hardware components such as VoIP switches, gateways, routers, telephone adaptors, and media servers.

All VoIP technology is immature.

We use boxes as adaptors to solve integration problems and make up for deficiencies in other components.

Modularity is especially important here —these are decisions we want to back out of, not embed deeply in code.

BOXES ARE INTERFACES TO WEB SERVICES