PROLOGUE

Building Serverless Solutions
with Azure and .NET

Introduction to Event-Driven Architecture

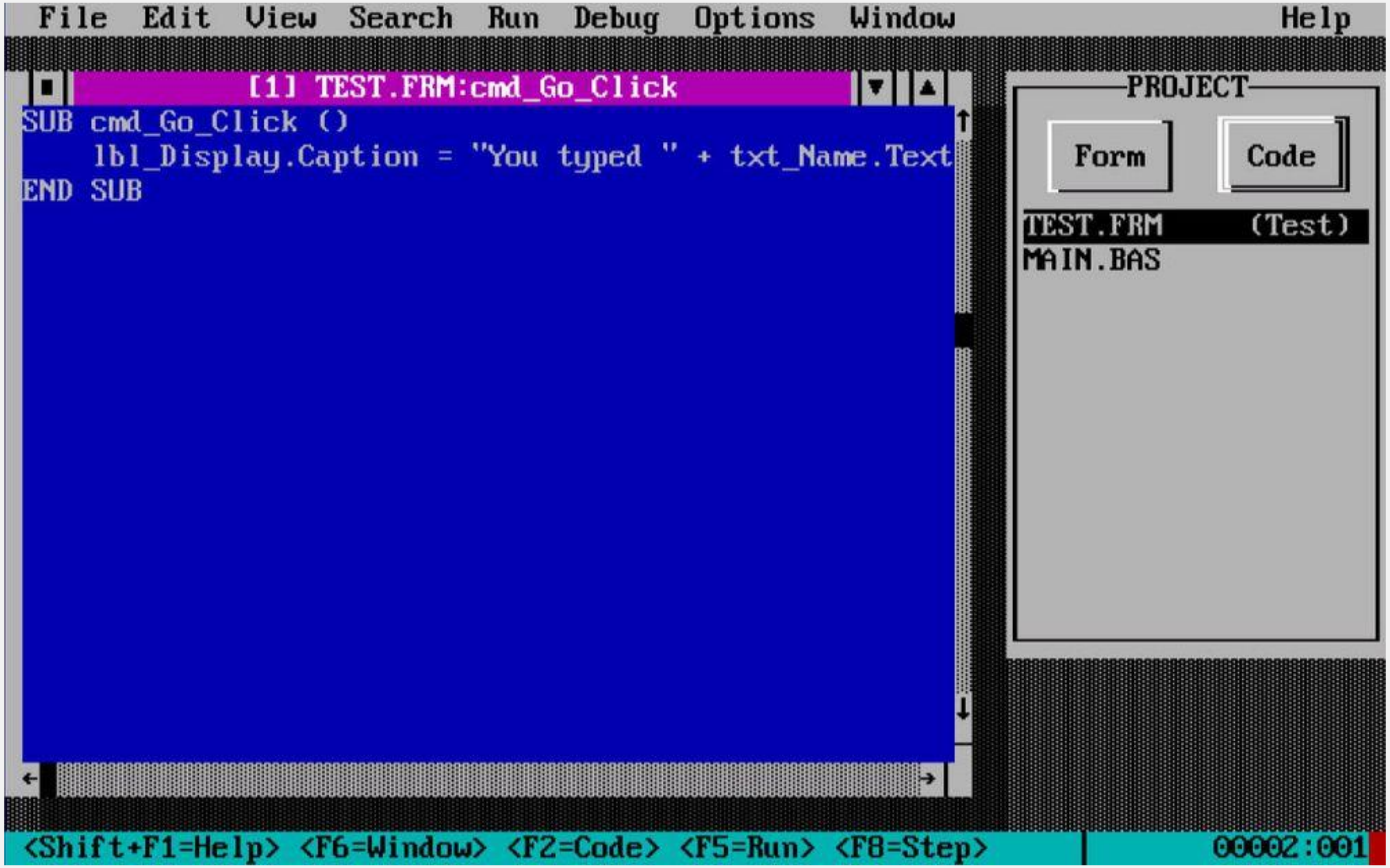
Understanding the Basics

What is Event-Driven Architecture?

> A software architecture pattern promoting the production, detection, consumption of, and reaction to **events**.

- Wikipedia -

What is Event-Driven Architecture?


File Edit View Search Run Debug Options Window Help
[1] TEST.FRM:cmd_Go_Click
PROJECT
Form
Code
TEST.FRM (Test)
MAIN.BAS
<Shift+F1=Help> <F6=Window> <F2=Code> <F5=Run> <F8=Step>
00002:001

```
SUB cmd_Go_Click ()
    lbl_Display.Caption = "You typed " + txt_Name.Text
END SUB
```

What is Event-Driven Architecture

Event-driven architecture (EDA) is a design paradigm in which a software component executes in response to receiving one or more event notifications.

EDA is more loosely coupled than client/server paradigm because the component that sends the notification doesn’t know the identity of the receiving components at the time of compiling

- Garner -

What is Event-Driven Architecture

“

Event-driven architecture (EDA) is a design paradigm in which a software component executes in response to receiving one or more event notifications.

EDA is more loosely coupled than client/server paradigm because the **component that sends the notification doesn’t know the identity of the receiving components** at the time of compiling

- Garner -

Key Components

Events

Key Components

Events

Event Emitters (Agents)

Key Components

Events

Event Emitters (Agents)

Event Consumers (Sinks)

Key Components

Events

Event Emitters (Agents)

Event Consumers (Sinks)

Event Channels

Key Components

Events

Event Emitters (Agents)

Event Consumers (Sinks)

Event Channels

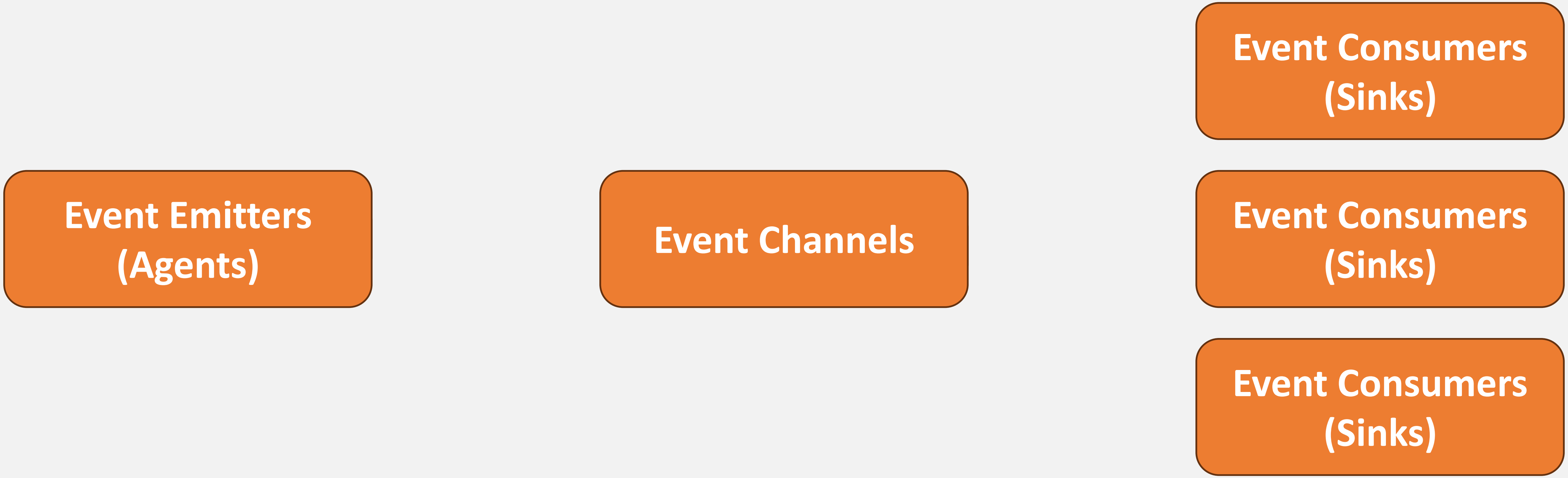
Key Components

Event Emitters (Agents)

Event Channels

Event Consumers (Sinks)

Key Components



Event Emitters (Agents)

Event Channels

Event Consumers (Sinks)

Event Consumers (Sinks)

Event Consumers (Sinks)

Benefits

Decoupling

Benefits

Decoupling

Scalability

Benefits

Decoupling

Scalability

Real-Time Responsiveness

Benefits

Decoupling

Scalability

Real-Time Responsiveness

Resilience

Benefits

Decoupling

Scalability

Real-Time Responsiveness

Resilience

Flexibility

Benefits

- Decoupling
- Scalability
- Real-Time Responsiveness
- Resilience
- Flexibility
- **Auditability**

Challenges

Complexity

Challenges

Complexity

Event Ordering

Challenges

Complexity

Event Ordering

Idempotency

Challenges

Complexity

Event Ordering

Idempotency

Event Schemas

Challenges

- Complexity
- Event Ordering
- Idempotency
- Event Schemas
- Testing

Challenges

- Complexity
- Event Ordering
- Idempotency
- Event Schemas
- Testing
- **Monitoring and Debugging**

Limitations

Guaranteed Delivery

Reliability

Limitations

Guaranteed Delivery

Reliability

At-Least-Once Delivery

Limitations

Guaranteed Delivery

Reliability

At-Least-Once Delivery

Ordering and Timing

Limitations

Guaranteed Delivery

- Reliability
- At-Least-Once Delivery
- Ordering and Timing
- **Monitoring and Error Handling**

Implementation Examples

Implementation Examples

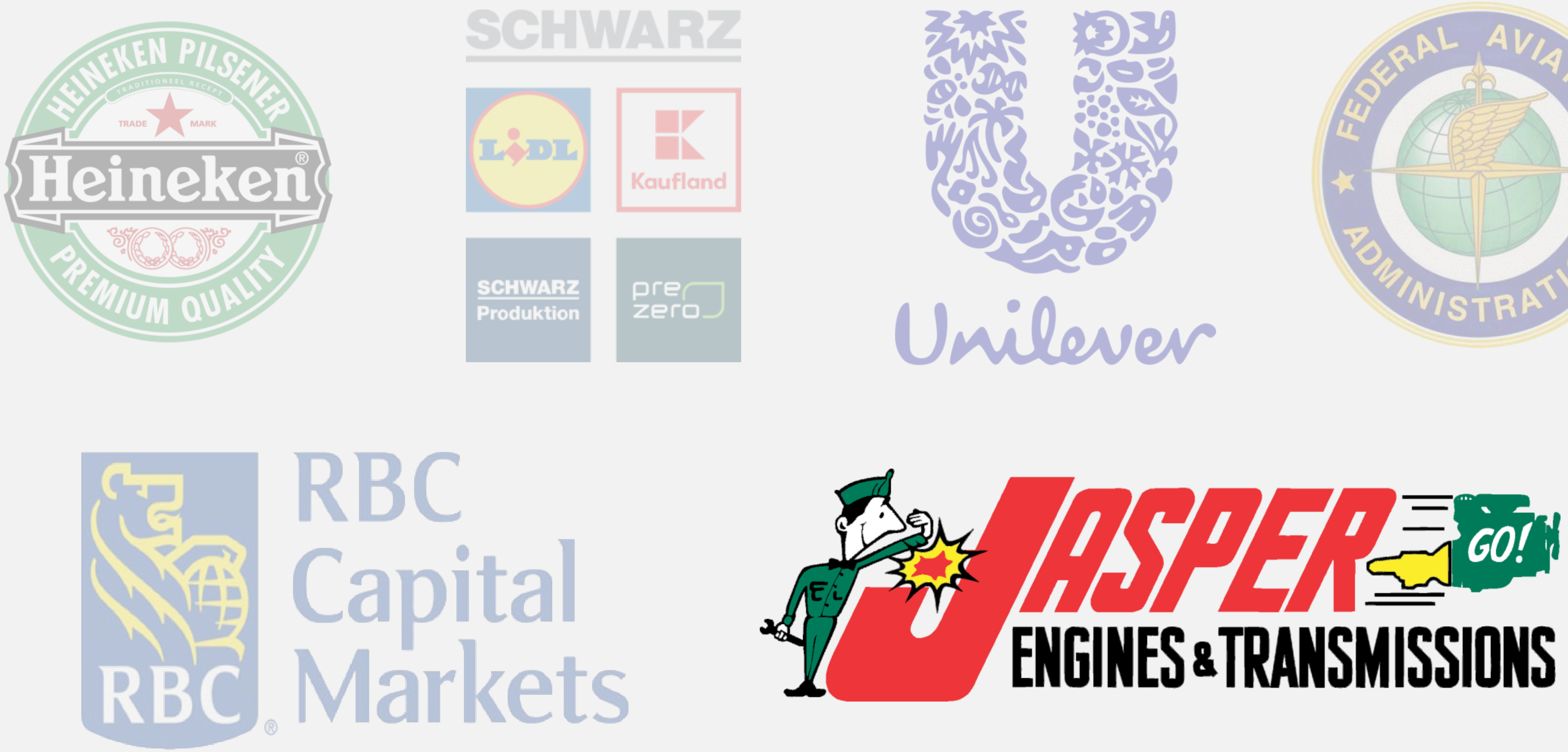
SCHWARZ
LIDL
Kaufland
SCHWARZ Produktion
pre zero

Implementation Examples

Implementation Examples

Implementation Examples

Implementation Examples



Conclusion

- Event-driven architecture offers several benefits but comes with its own challenges.
- It is particularly well-suited for real-time, asynchronous, and distributed applications.

Introduction to Serverless

Beyond Traditional Infrastructure