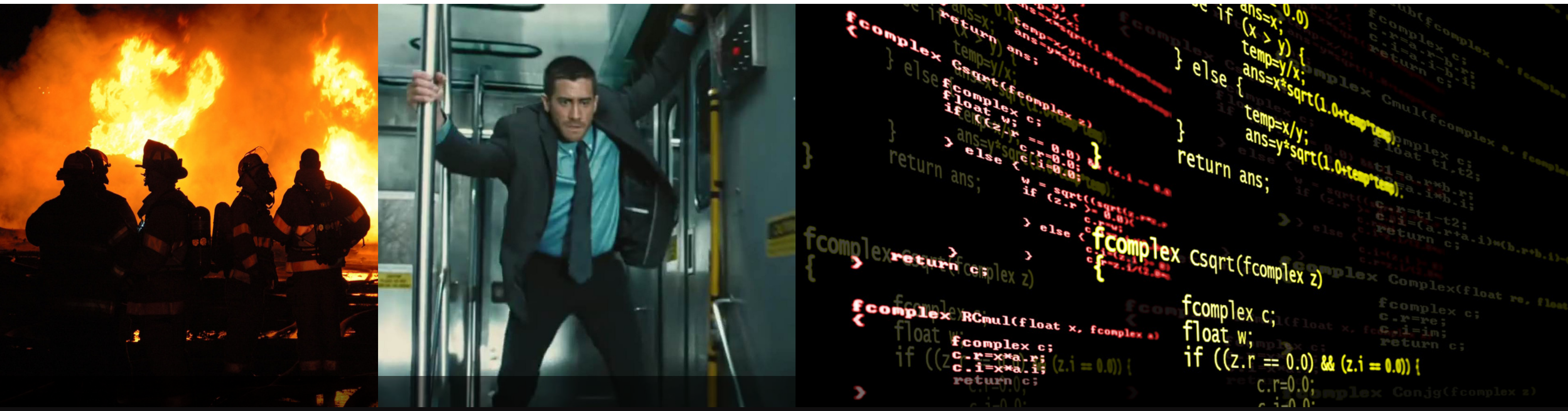


Proactive Computing:
Changing the Future

SPECIAL REPORT

RTInsights.com

Accelerate Your Business with Real-Time Insights

AUTHOR:

Dr. Opher Etzion

Professor and Head of the Technological Empowerment Institute

Yezreel Valley College, Israel

and

RTInsights International Technology Editor

Introduction: Future Scenarios

In the movie "Source Code" with Jake Gyllenhaal, the plot shifts from reacting to an already occurred train explosion to changing reality to prevent the explosion from happening in the first place. While changing history is still within the domain of science fiction, changing reality to avoid events that have not happened is useful and feasible.

Imagine a world where it is possible to prevent problems or capitalize on opportunities before they even occur—for example, prevent traffic congestion, optimize power consumption based on weather and electric grid demands, or know which roads and schools to close to prevent further damage immediately after an earthquake.

Proactive Event-Driven Computing is a new paradigm, where a decision is autonomously triggered by forecasting future events—whether they are problems or opportunities—instead of reacting once they happen. The decisions and actions are often real time in the sense that they are taken under time constraints and require the exploitation of extremely large amounts of historical and streaming data. Although the idea of proactive computing appears simple, it is still an emerging area that has not reached maturity in mainstream computing.

We use the term proactive behavior to designate changing future reality.
Some examples include:

- **Proactive condition-based maintenance (CBM)**: The target event is the failure of a piece of critical equipment (e.g., drilling equipment used by an oil company). This situation can be forecasted based on observations of events related to this equipment or the surrounding environment (e.g. weather). An action that completely eliminates this target event is to take the equipment down for full maintenance; other less drastic and less costly actions (such as lubrication of metal parts) only reduce the probability of failure.

- **Proactive elimination of gridlock in mining roads:** Some roads within mining facilities (e.g. copper mines) are wide enough for just a single truck; drivers use simple communication mechanisms to assign the right of way. It does happen however that two trucks are coming from different directions at the same time (the target event), in which case the operation to resolve the gridlock is very expensive and risky. Avoiding such situations before they occur is extremely beneficial.

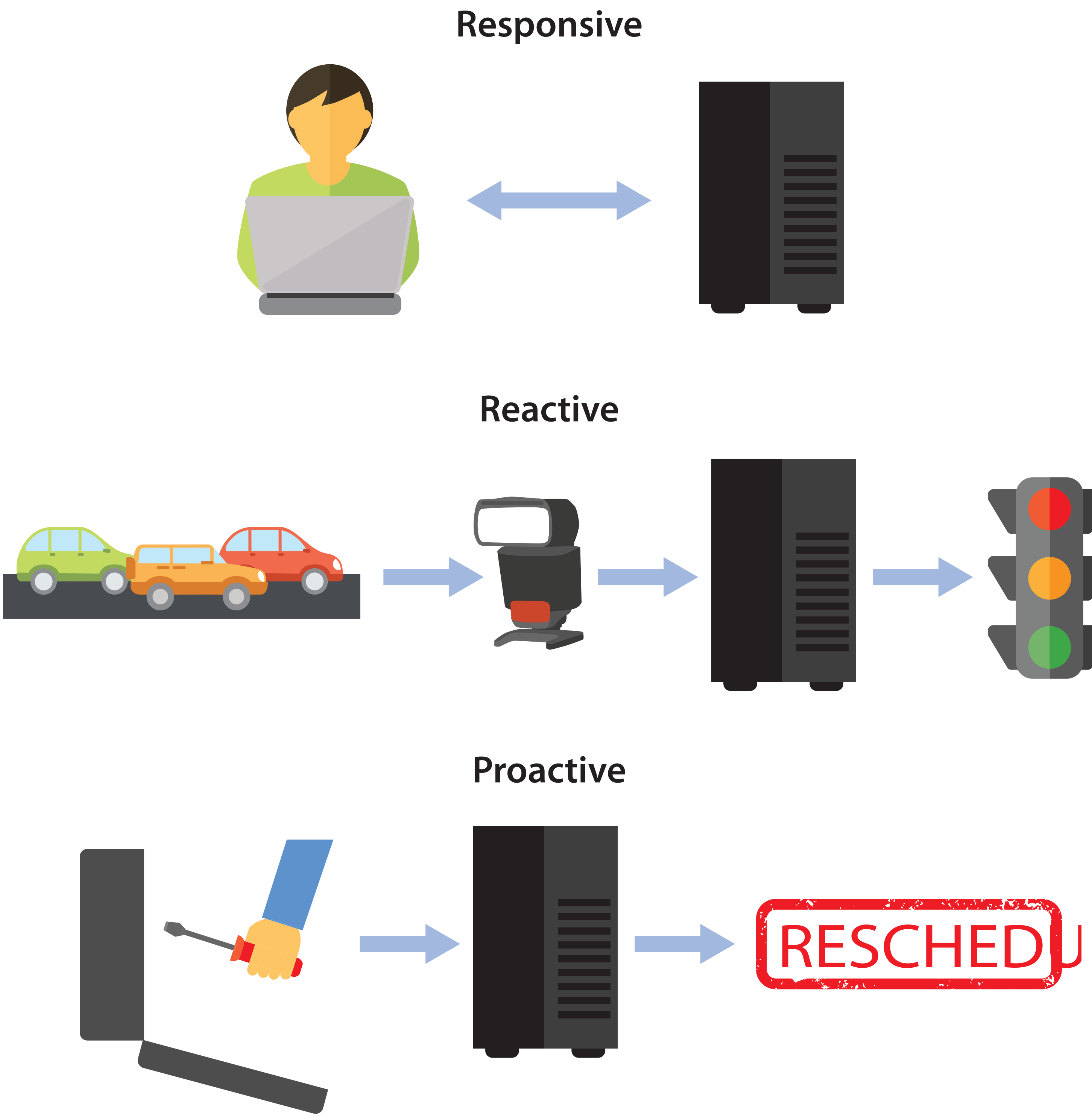
- **Proactive response to a Denial of Service (DoS) attack**: DoS attacks are well-known threats to various computer services. There is rich literature on early detection of DoS attacks and results on the use of event patterns as early indicators. After detecting that a service is under attack, there are various negative events that can be predicted (the target events) depending on the particular service, such as unavailability of the service, server crashes, or other kinds of network failure. Consequentially, there are various responses that can be applied if the attack is detected early (such as adding computing resources, shutting down less important services, and dropping requests either selectively or arbitrarily). In the proactive paradigm, the future risks are predicted (using probability) according to the characteristics of current events, and weighed against the cost of the preventive measures.

From Responsive to Proactive

There is a recent evolution from the responsive approach of traditional computing, in which the computer responds to an explicit request by the user ("request-response") to the reactive approach in which systems react to events. The proactive approach, illustrated in Figure 1, is the next phase in the evolution. Proactive event-based computing is different from predictive analytics by the fact that it is event-driven, typically geared towards operational decisions, and operates in real time, while predictive analysis is typically retrospective and mainly geared towards strategic decisions.

FIGURE 1:

Evolution of computing paradigms

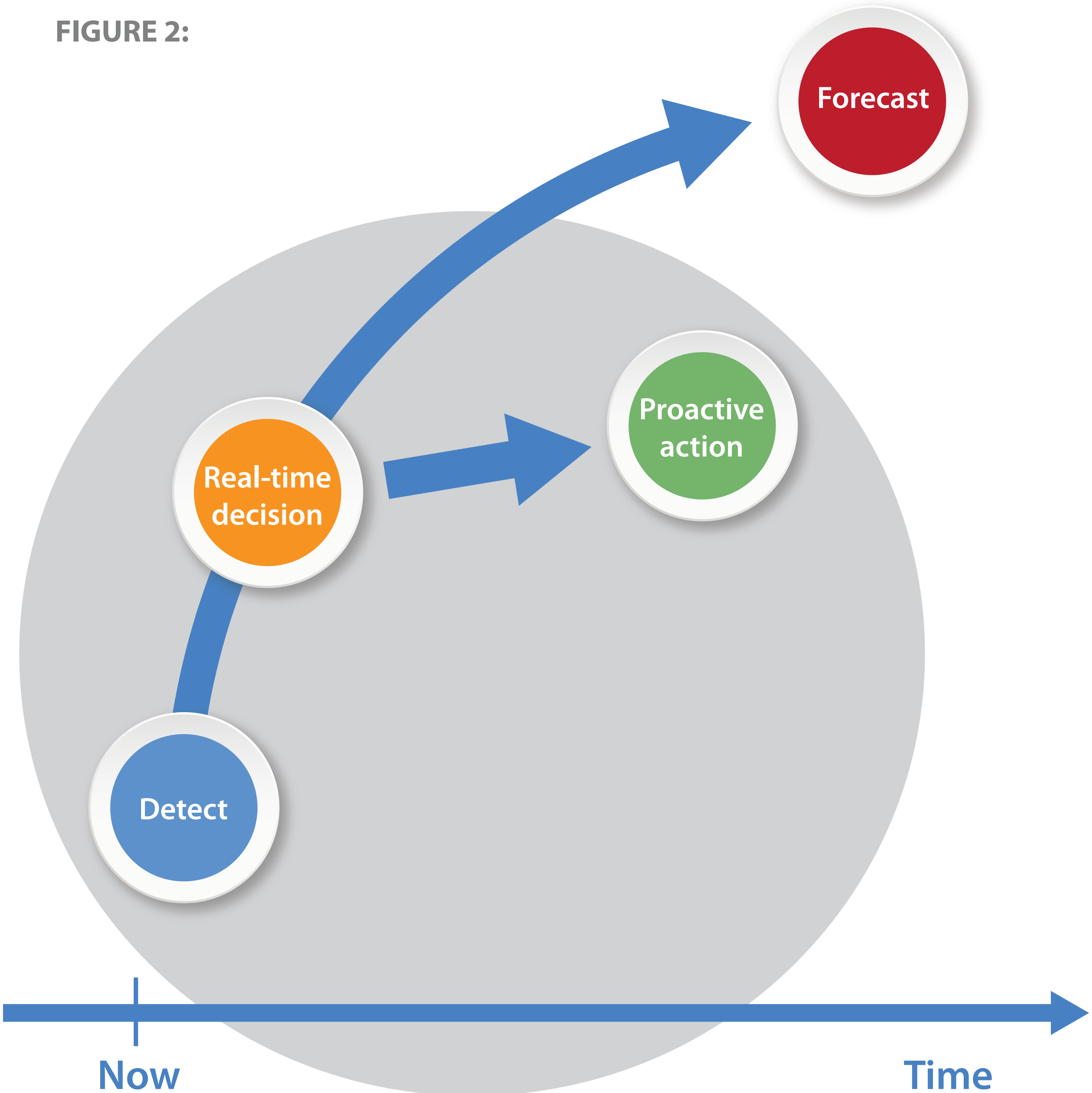


The Proactive Event-Driven Principle

The circle in Figure 2 shows the space of admissible states, and the proactive pattern consists of four states:

- Detect: Monitoring the universe; a detection of the current indicators.
- Forecast: The current indicators are used to forecast that the system is going to a state outside the admissible state in the future if nothing changes.
- Real-time decision: A decision about the best way to eliminate or mitigate the problem and stay within the admissible states.
- Proactive action: An action taken in the wake of the decision.

FIGURE 2:



Let's consider the following scenario:

Francois has to travel from Luxembourg to Tromso in Norway to participate in a climate-change conference. The travel agent books him a two-leg flight through Copenhagen to Oslo. Due to heavy snow in Oslo, the airport was shut. Unfortunately, Francois discovered this only after landing in Copenhagen. There were no alternative ways to go from Copenhagen in the short run and Francois was stranded in Copenhagen and did not get in time for his talk. If we use a proactive system, the scenario will change as follows:

Detect

At 09:00 the proactive system detects warnings about weather conditions in the route and in alternative routes

Forecast

The proactive system forecasts that the Oslo airport will be closed between 11:00 a.m. and 12:00 p.m. with probability of 0.5

Decide

The proactive system decides that the optimal plan is to fly to Frankfurt and assess the situation. From Frankfurt there are flights to both Oslo and Bergen.

Act

Francois arrives at Frankfurt and flies to Tromso via Bergen. He gets there in time and gives his talk – and this results in a huge grant…

Some other proactive scenarios are illustrated on the next pages:

FIGURE 3:

Scenario 1: Disaster Management

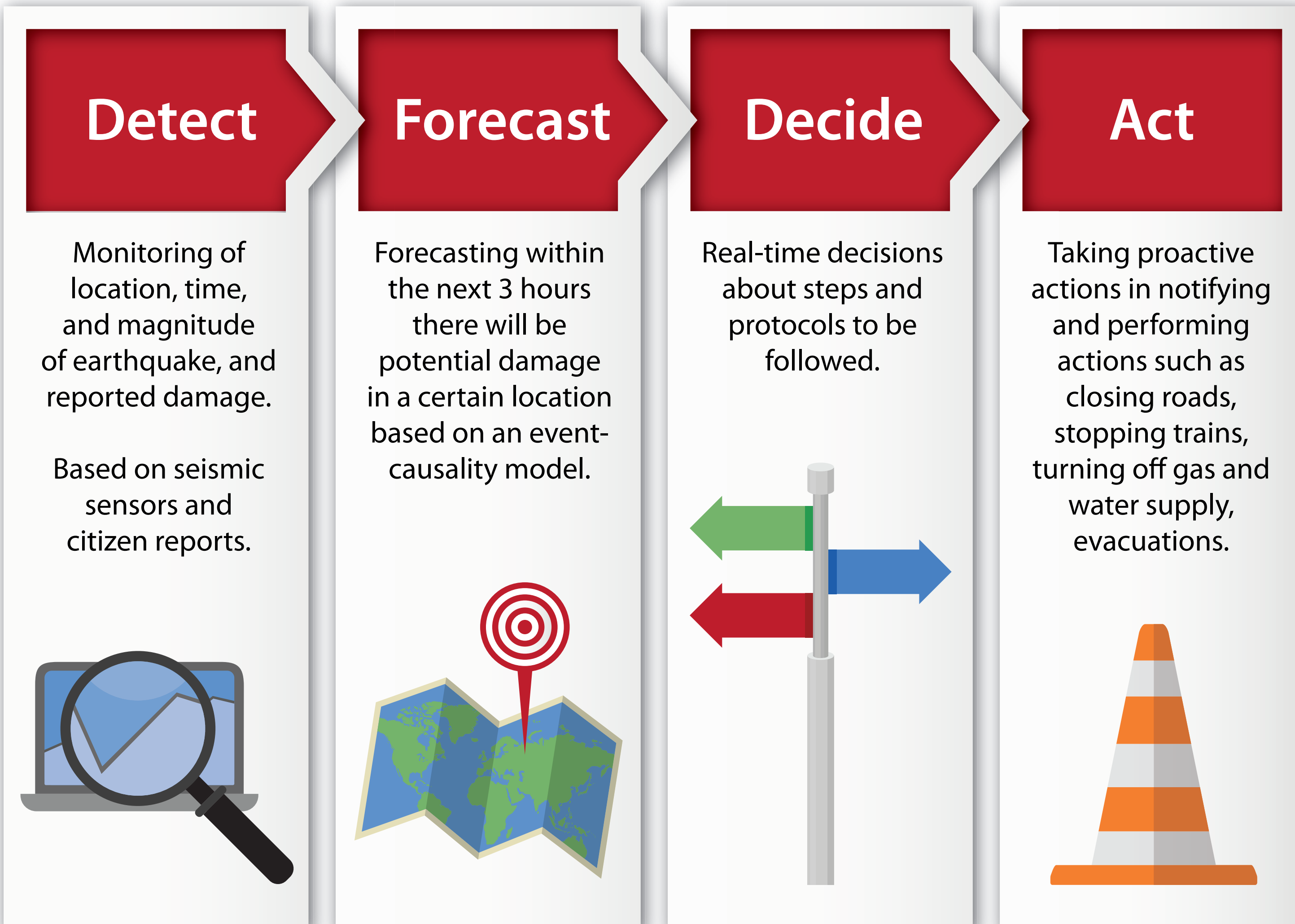
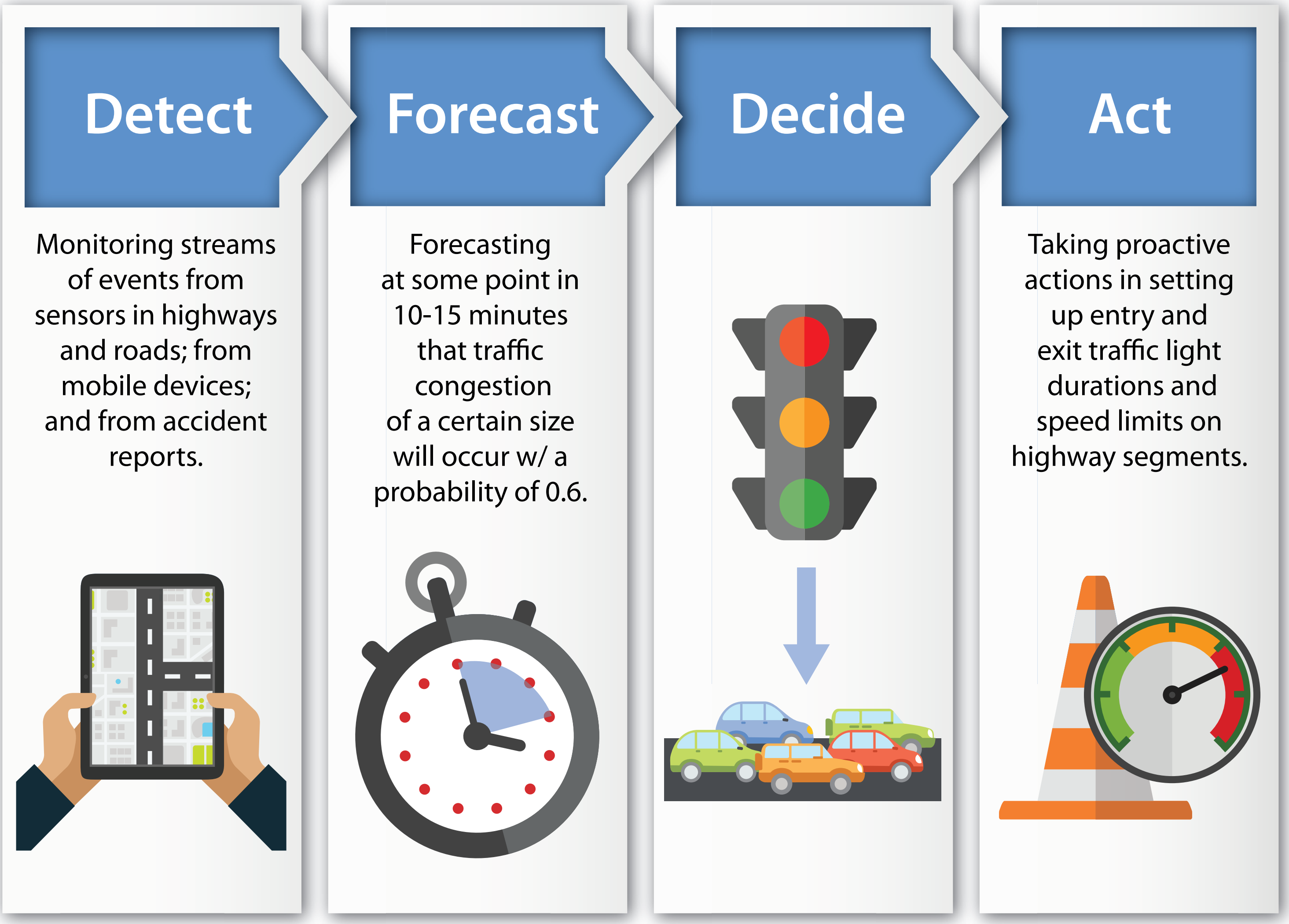


FIGURE 4:

Scenario 2: Road Management

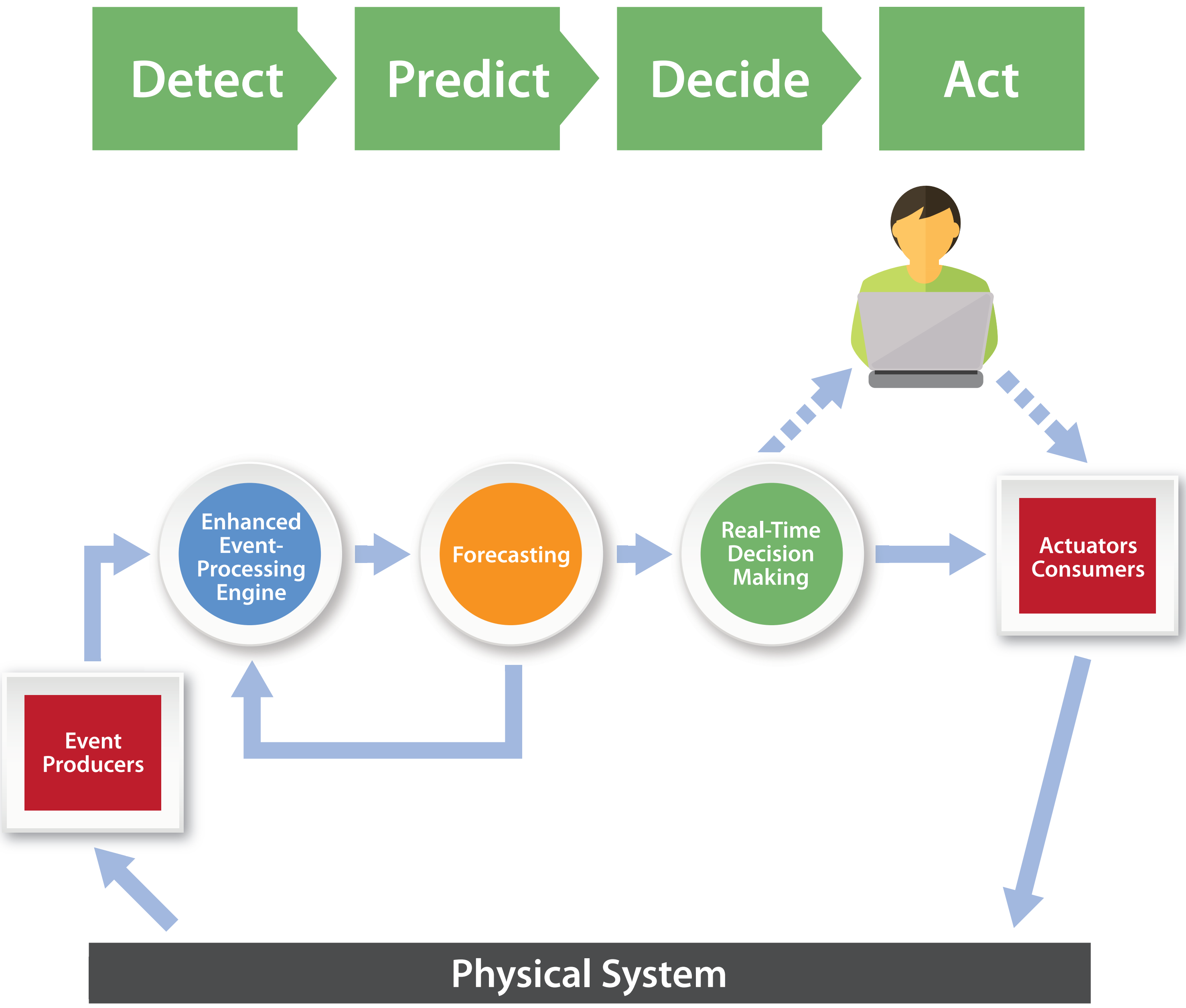


Use pattern of proactive event-driven systems

Figure 5 shows an architectural view of an event-driven system. An enhanced event-processing system able to handle future events is used for the detect phase, the forecasting and decisions are used as event-driven artifacts. The event consumers also include actuators.

This architecture also demonstrates the difficulty in building such systems, due to the fact that it uses a blend of technology, with a need to integrate not only software artifacts but also programming models. The challenge of building a constructive unified programming model is still emerging.

FIGURE 5:



Looking Ahead

A proactive event-driven computing paradigm has the potential to change life in many areas, moving from reacting to events that already happened to eliminating events from happening. This approach is taken today in computational biology, where sensors and actuators aim to provide insights that can fight diseases before they inflict damage,and even fight the aging mechanism of the human body. We shall hear about this paradigm much more within the coming years.

-Dr. Opher Etzion is a professor of Information Systems and Head of the Technological Empowerment Institute in Yezreel Valley College in Israel. Dr. Etzion has served in various prominent roles in almost two decades at IBM, most recently as Chief Scientist of Event Processing at the IBM Haifa Research Lab. Dr. Etzion is also the founding chair of the Event Processing Technical Society (EPTS) and an adjunct professor at the Technion Israel Institute of Technology. He is the co-author with Peter Niblett of the book Event Processing in Action.

Photo credits: Cover art, train, Jake Gyllenhaal: "Source Code" official trailer. Figures 1-5 adapted from IBM Hafia Research Lab presentation.