

# Designing Interactive Systems 2

## Lecture 1: Introduction, History, Design Space of Input Devices

Prof. Dr. Jan Borchers  
Media Computing Group  
RWTH Aachen University

[hci.rwth-aachen.de/dis2](http://hci.rwth-aachen.de/dis2)



**RWTH**AACHEN  
UNIVERSITY

# Class Syllabus

- Part 1  
**Key Concepts**
- Part 2  
**Usage and Design of UI Toolkits and Design Systems**
- Part 3  
**UIs Beyond the Desktop**
- Part 4  
**Prototyping Process**



# Administrivia

- Format: V3/Ü2
- 6 Credit points
- Class times
  - **Lecture** on Wednesdays (8:30 — 11:00), Room 2222
  - **Lab** on Mondays (14:30 — 16:00), Room 2222





# Team



**Prof. Dr.  
Jan Borchers**



**Sebastian  
Hueber**

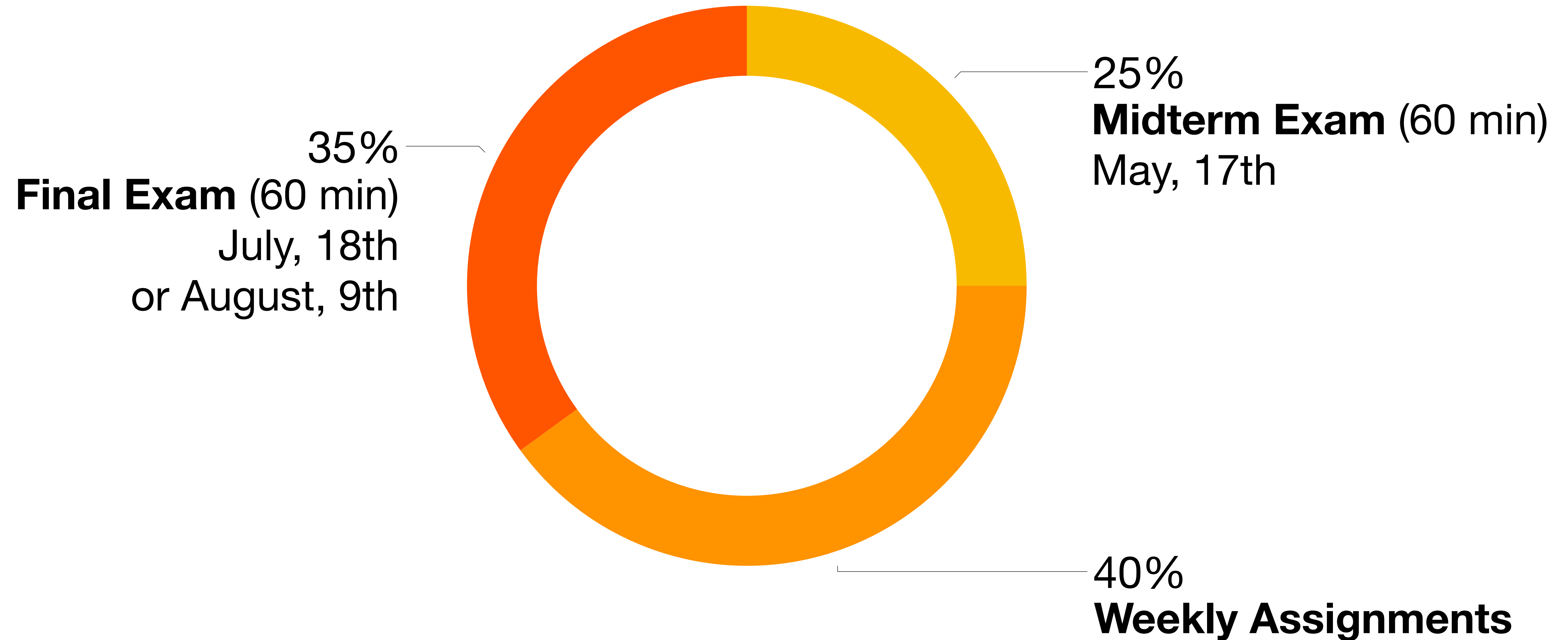
[hueber@cs.rwth-aachen.de](mailto:hueber@cs.rwth-aachen.de)  
E-Mail Subject: [DIS2]



**Anke  
Brocker**

[brocker@cs.rwth-aachen.de](mailto:brocker@cs.rwth-aachen.de)  
E-Mail Subject: [DIS2]

# Your Final Grade



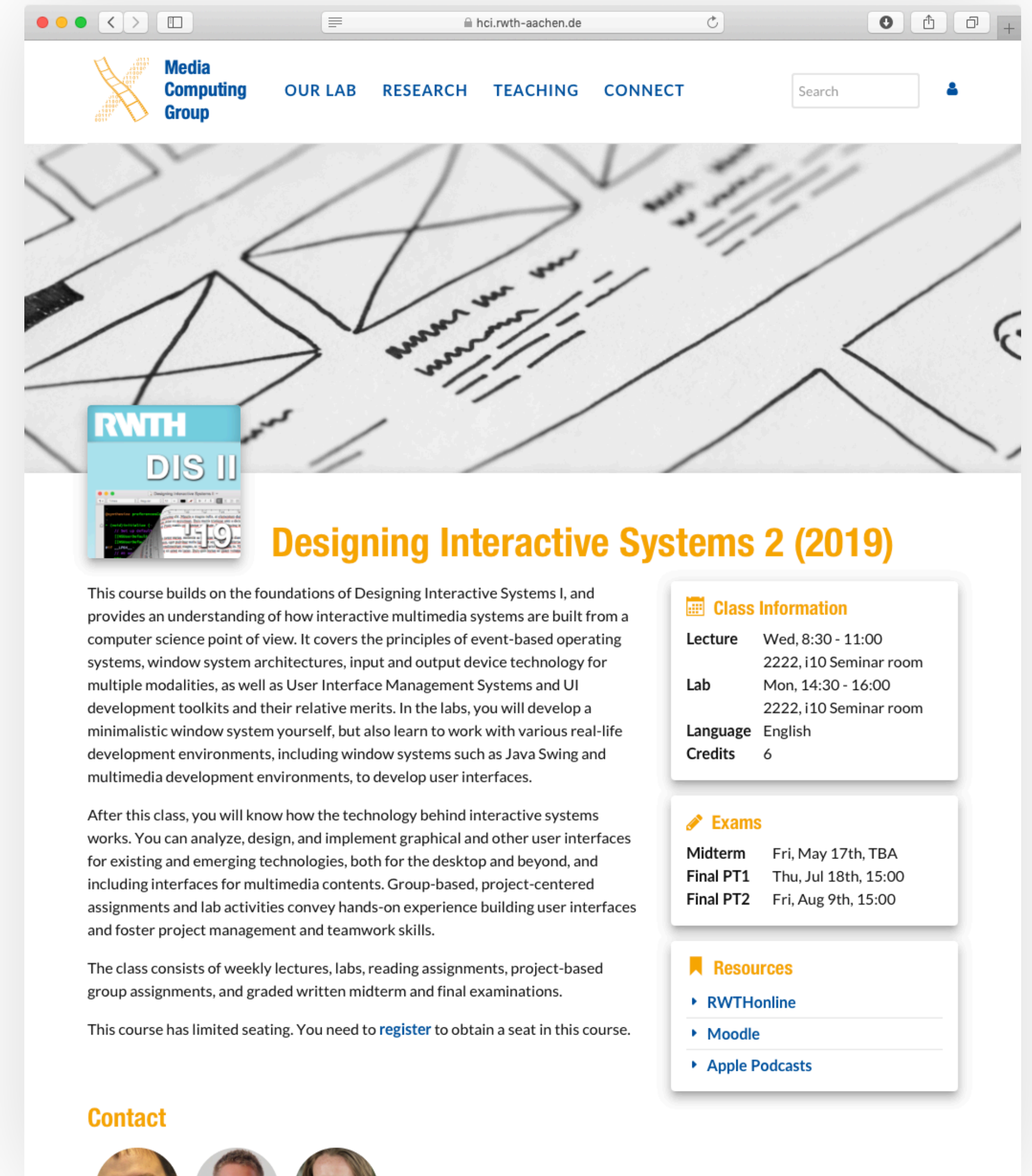
# Weekly Assignments

- We have a strict grading policy:
  - **Late submissions** will be graded 5.0 without feedback
  - **Team size** is 2 students (3 only by permission). If you hand in a solution without a team partner: 5.0 without feedback
  - If your code does **not compile**: 5.0 without feedback
- For some assignments you will need a **Mac**
  - No Mac? Visit <http://www-rbi.informatik.rwth-aachen.de/Pool+Helpdesk/>
- Submission via **Moodle**



# Website

- All information about this course can be found online
- [hci.rwth-aachen.de/dis2](https://hci.rwth-aachen.de/dis2)



The screenshot shows the website for the Media Computing Group at RWTH Aachen University. The header includes the group's logo and navigation links: OUR LAB, RESEARCH, TEACHING, and CONNECT. A search bar is also present. The main content area features a large background image of a hand-drawn diagram. Below this, a blue box with the text 'RWTH DIS II' is visible. The course title 'Designing Interactive Systems 2 (2019)' is prominently displayed in orange. The course description states that it builds on the foundations of Designing Interactive Systems I, covering principles of event-based operating systems, window system architectures, input and output device technology, and User Interface Management Systems. It also mentions that students will develop a minimalistic window system and learn to work with various real-life development environments. The course is described as group-based and project-centered, with hands-on experience building user interfaces and fostering project management and teamwork skills. The class consists of weekly lectures, labs, reading assignments, project-based group assignments, and graded written midterm and final examinations. A note mentions limited seating and the need to register. The course information table lists the lecture and lab times, language, and credits. The exams table lists the midterm and final exam dates and times. The resources section includes links to RWTHOnline, Moodle, and Apple Podcasts. The contact section at the bottom shows three profile pictures.

Media Computing Group

OUR LAB RESEARCH TEACHING CONNECT

Search

RWTH DIS II

## Designing Interactive Systems 2 (2019)

This course builds on the foundations of Designing Interactive Systems I, and provides an understanding of how interactive multimedia systems are built from a computer science point of view. It covers the principles of event-based operating systems, window system architectures, input and output device technology for multiple modalities, as well as User Interface Management Systems and UI development toolkits and their relative merits. In the labs, you will develop a minimalistic window system yourself, but also learn to work with various real-life development environments, including window systems such as Java Swing and multimedia development environments, to develop user interfaces.

After this class, you will know how the technology behind interactive systems works. You can analyze, design, and implement graphical and other user interfaces for existing and emerging technologies, both for the desktop and beyond, and including interfaces for multimedia contents. Group-based, project-centered assignments and lab activities convey hands-on experience building user interfaces and foster project management and teamwork skills.

The class consists of weekly lectures, labs, reading assignments, project-based group assignments, and graded written midterm and final examinations.

This course has limited seating. You need to [register](#) to obtain a seat in this course.

### Class Information

Lecture	Wed, 8:30 - 11:00 2222, i10 Seminar room
Lab	Mon, 14:30 - 16:00 2222, i10 Seminar room
Language	English
Credits	6


### Exams

Midterm	Fri, May 17th, TBA
Final PT1	Thu, Jul 18th, 15:00
Final PT2	Fri, Aug 9th, 15:00

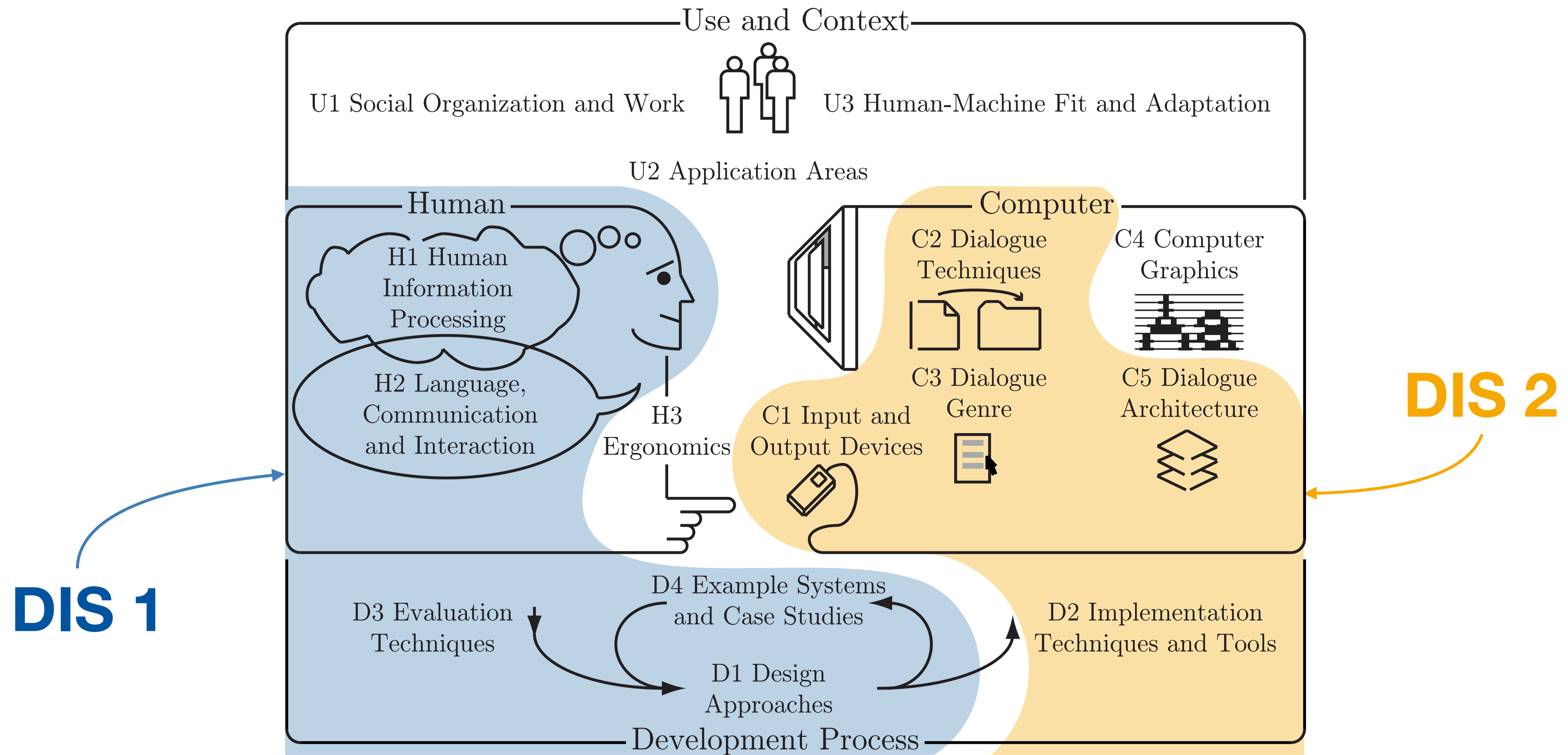
### Resources

- [RWTHOnline](#)
- [Moodle](#)
- [Apple Podcasts](#)

### Contact

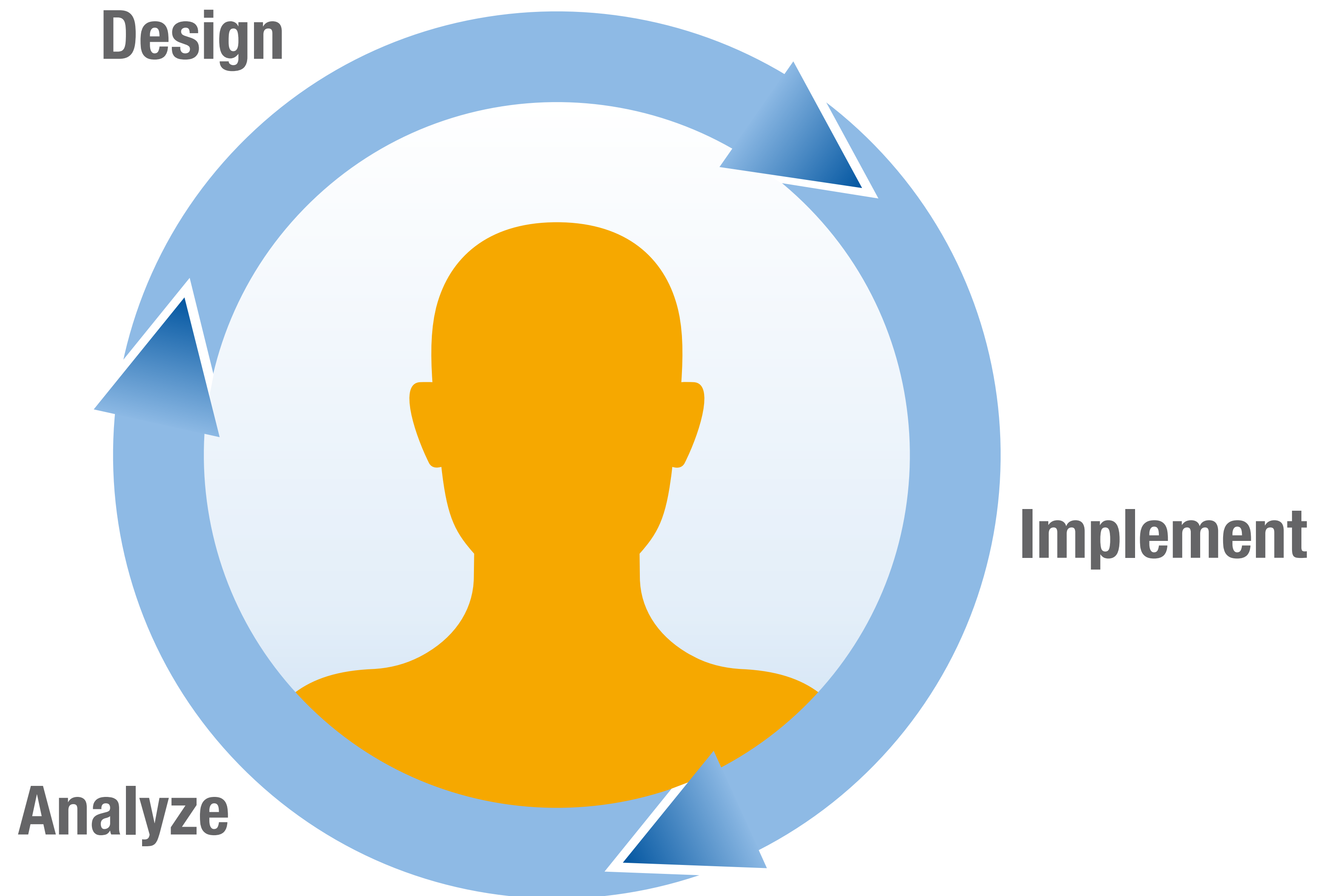


# How DIS1 and DIS2 Cover HCI





# DIA Cycle

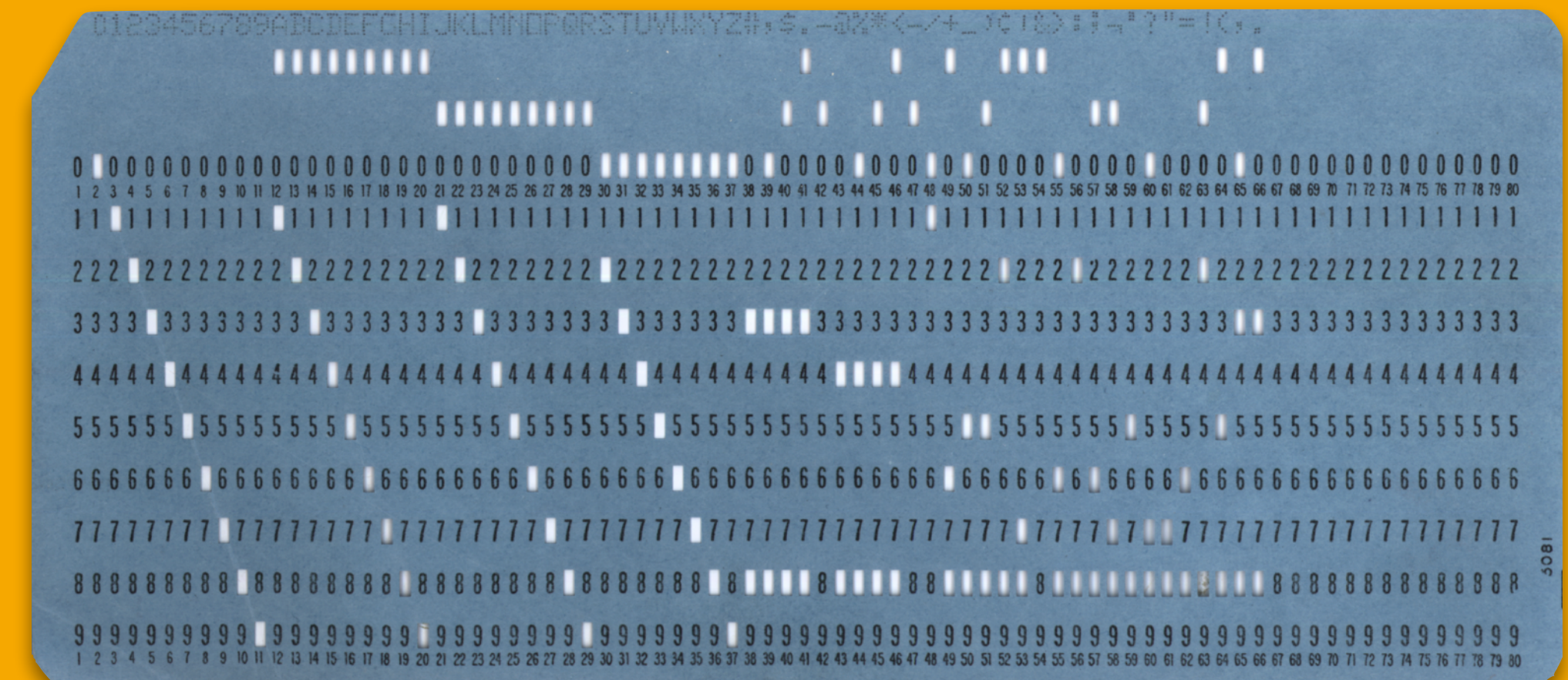
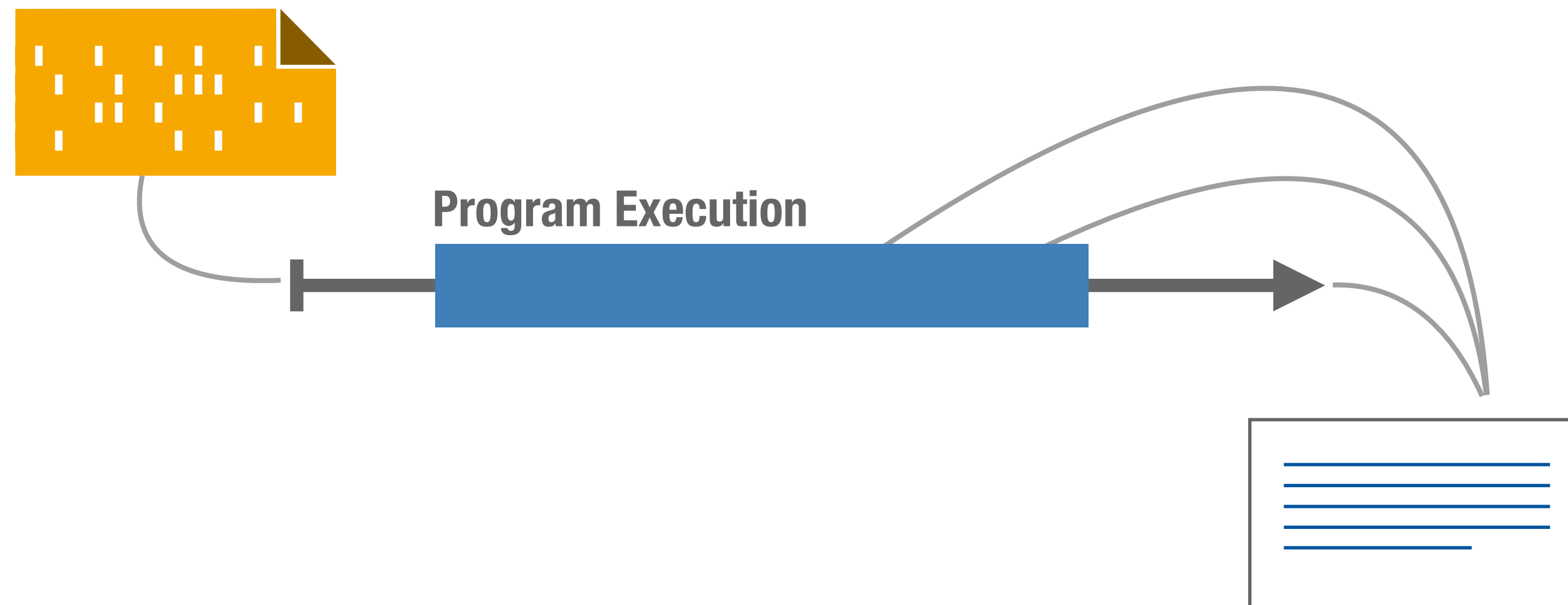


## CHAPTER 1

# History of User Interface Programming Paradigms

# Batch Processing

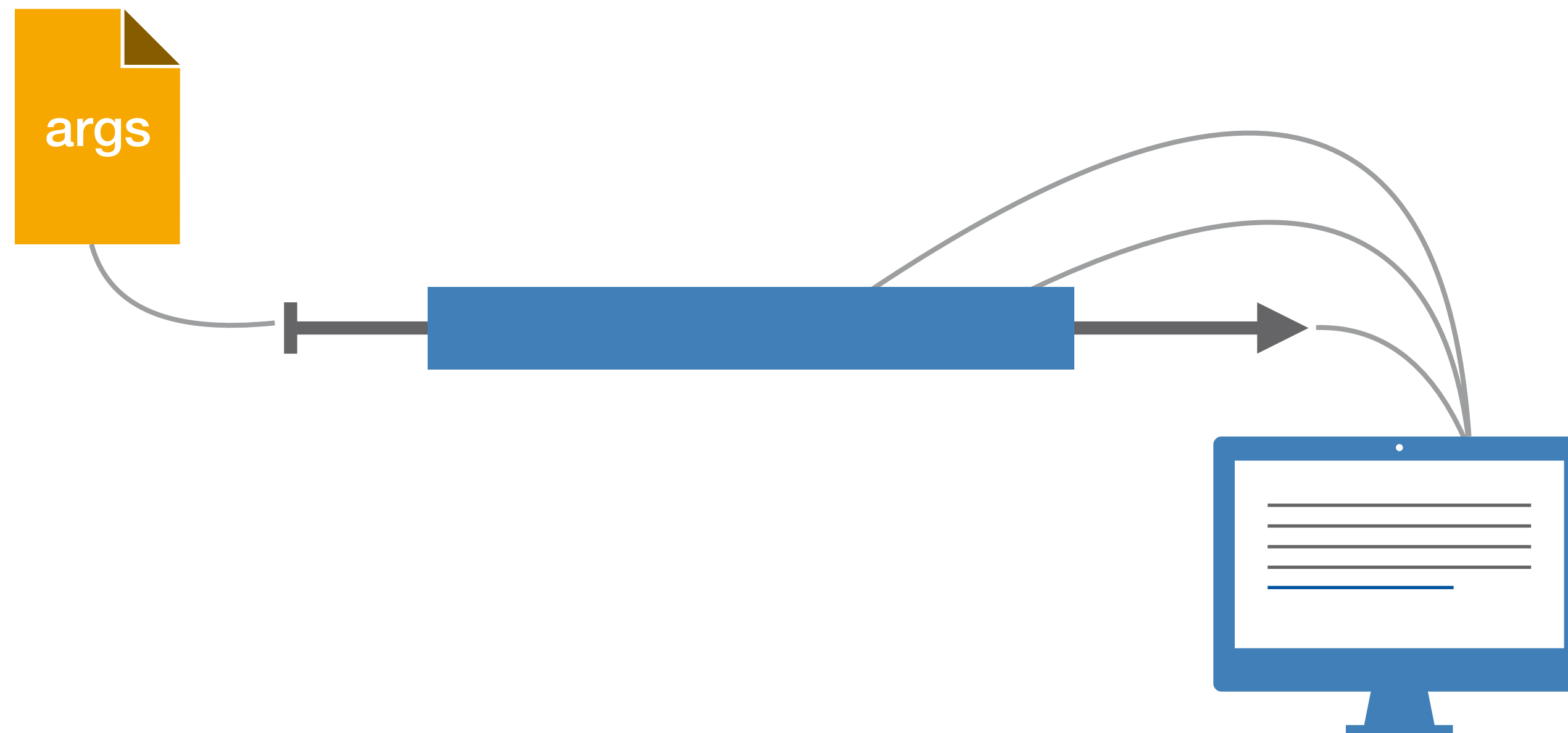
- Prepare data on punch cards
- Wait for result as printout offline





# Time-sharing Systems

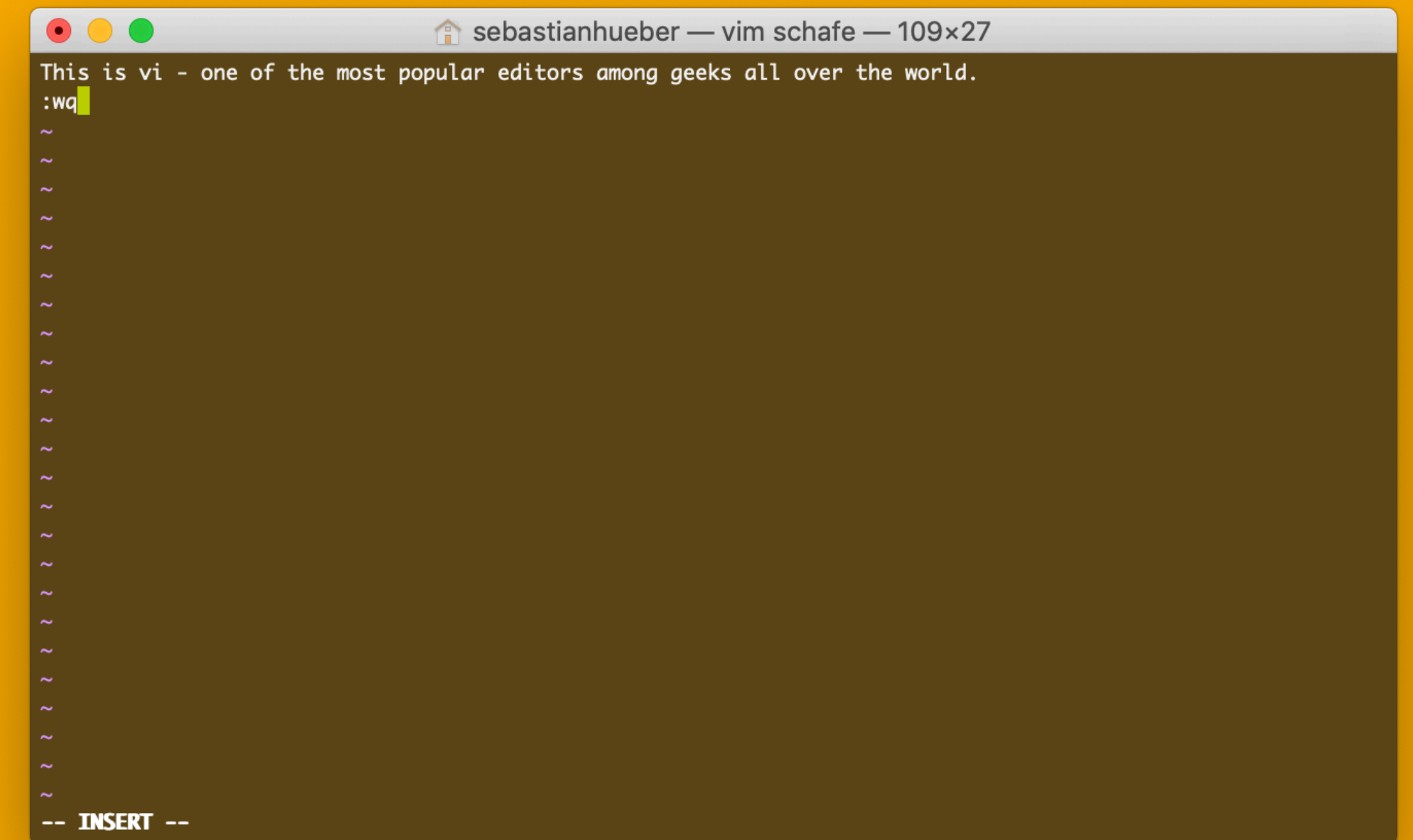
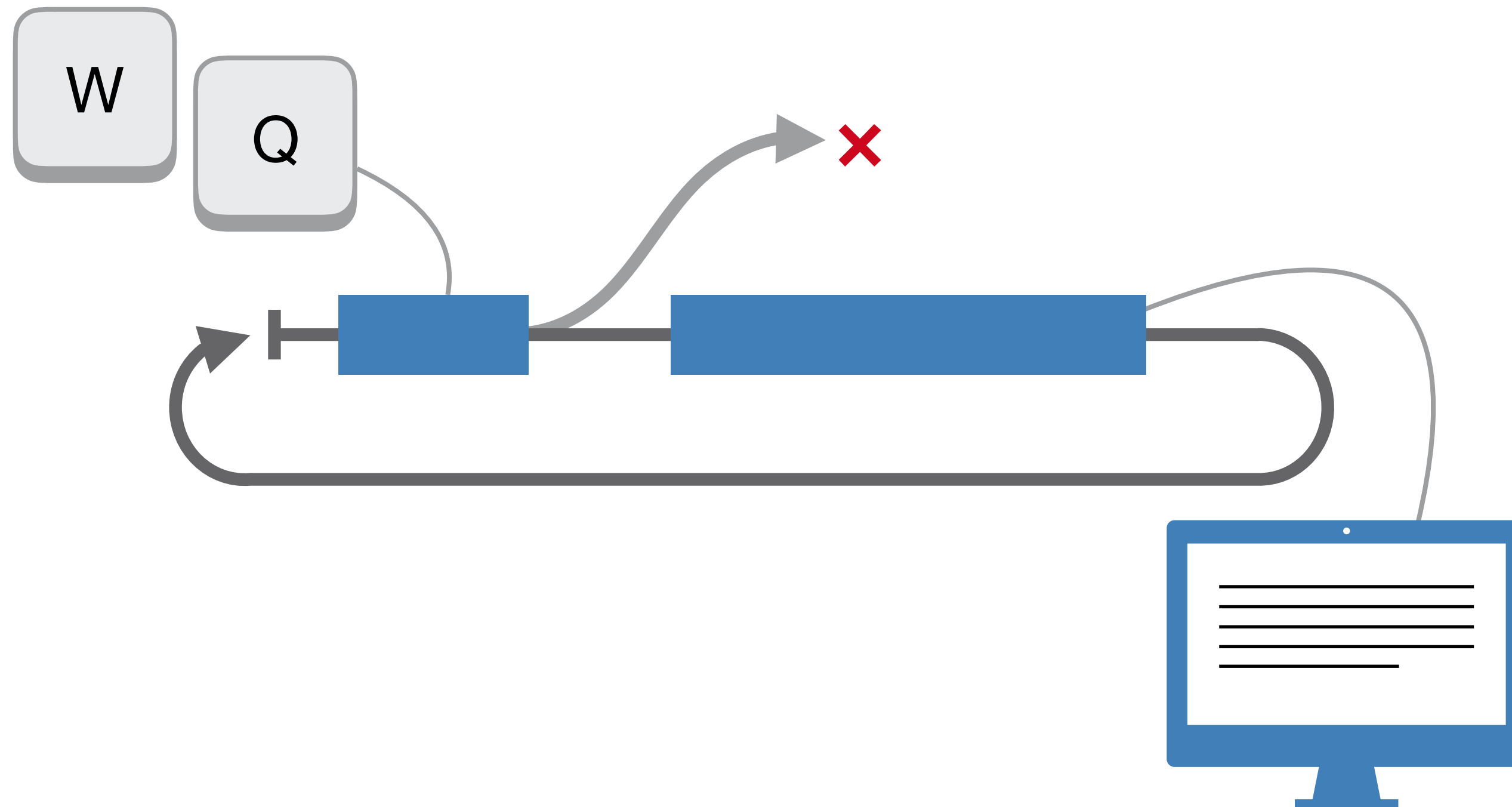
- Command-line based interaction
- Shorter turnaround (per-line)





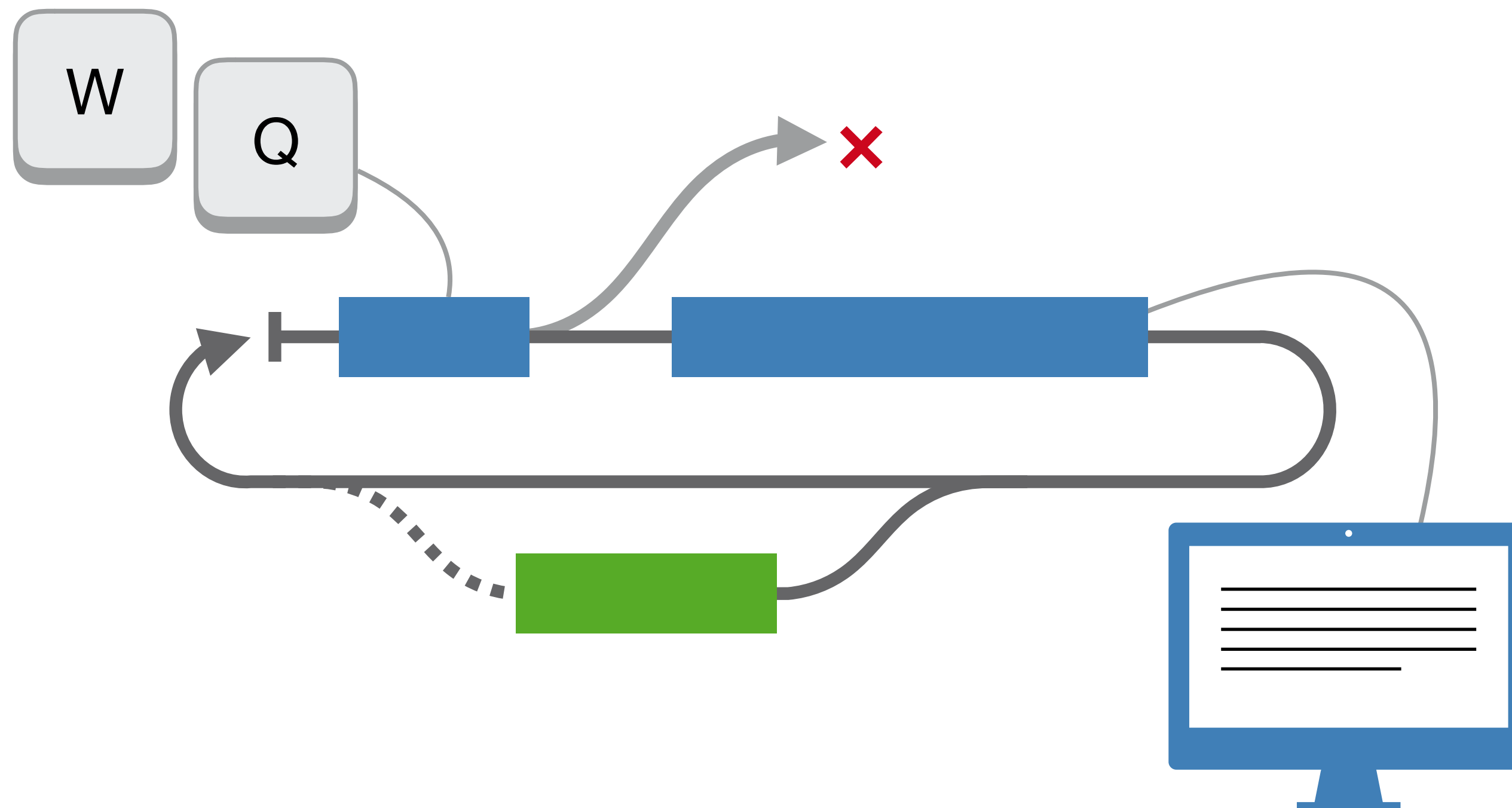
# Full-screen textual UIs

- Turnaround per character
- Interaction starts to feel “real-time”



# Menu-based Systems

- Discover functionalities instead of memorizing them
- **Threading** becomes important



```

GNU nano 2.0.6      File: /Library/Python/2.7/site-packages/vboxapi/__init__.py

# -*- coding: utf-8 -*-
# $Id: vboxapi.py 101359 2015-06-30 22:28:00Z bird $
"""
VirtualBox Python API Glue.
"""

__copyright__ = \
"""
Copyright (C) 2009-2015 Oracle Corporation

This file is part of VirtualBox Open Source Edition (OSE), as
available from http://www.virtualbox.org. This file is free software;
you can redistribute it and/or modify it under the terms of the GNU
General Public License (GPL) as published by the Free Software
Foundation, in version 2 as it comes in the "COPYING" file of the
VirtualBox OSE distribution. VirtualBox OSE is distributed in the
hope that it will be useful, but WITHOUT ANY WARRANTY of any kind.
"""

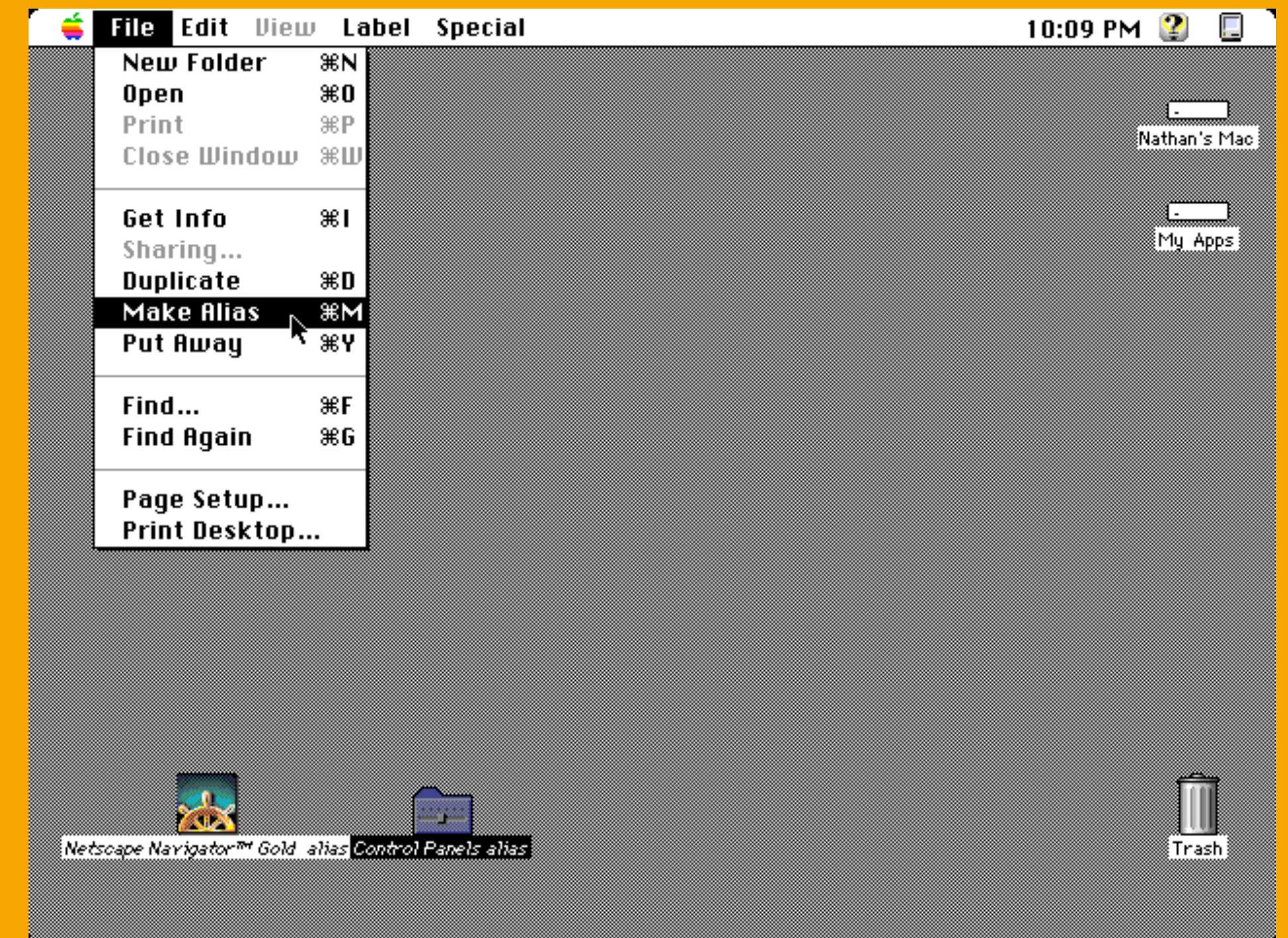
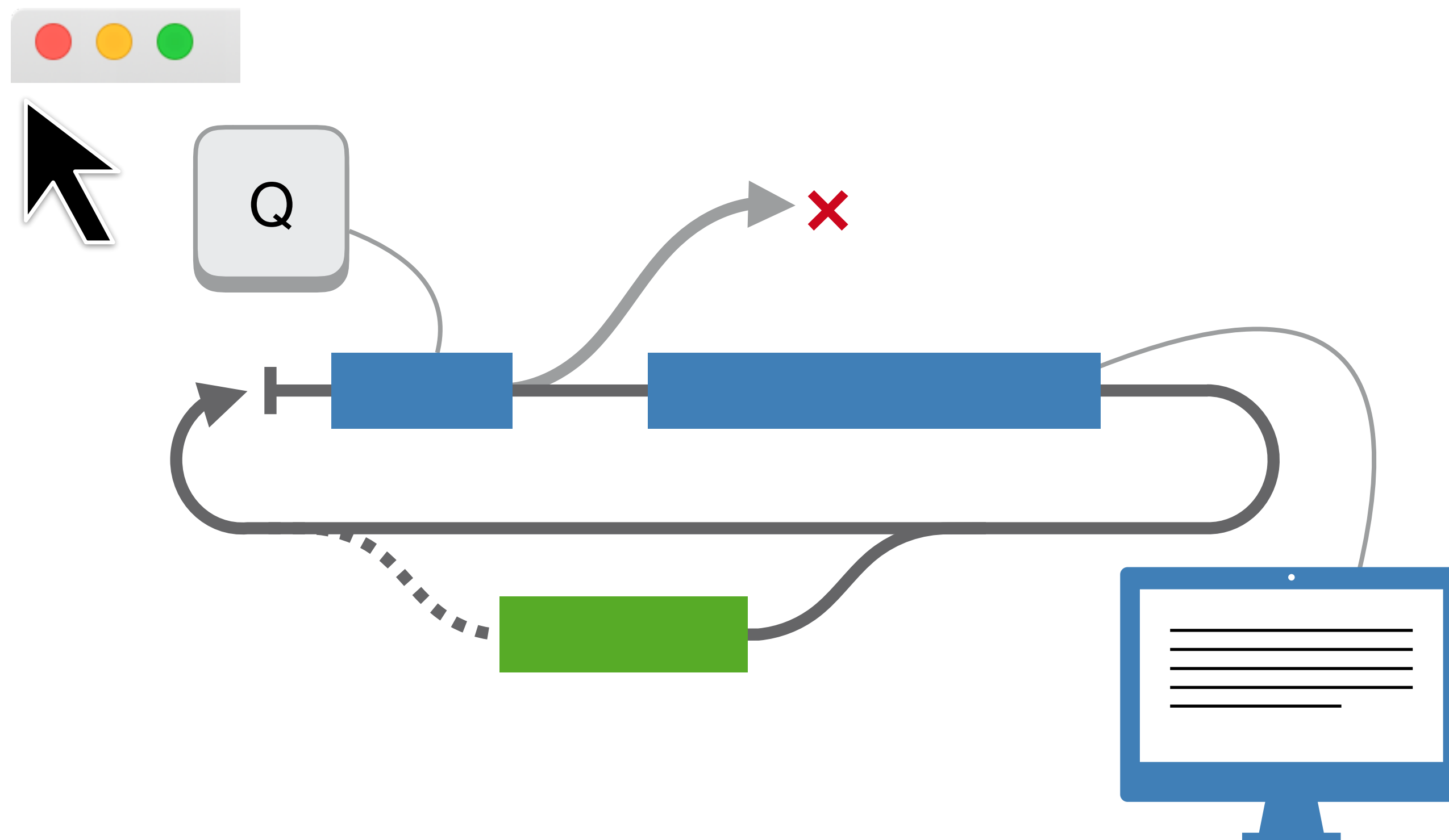
__version__ = "$Revision: 101359 $"

# Note! To set Python bitness on OSX use 'export VERSIONER_PYTHON_PREFER_32_BIT=yes'

```

# Graphical User Interface

- Event-based program structure
- Pointing devices in addition to keyboard





## CHAPTER 2

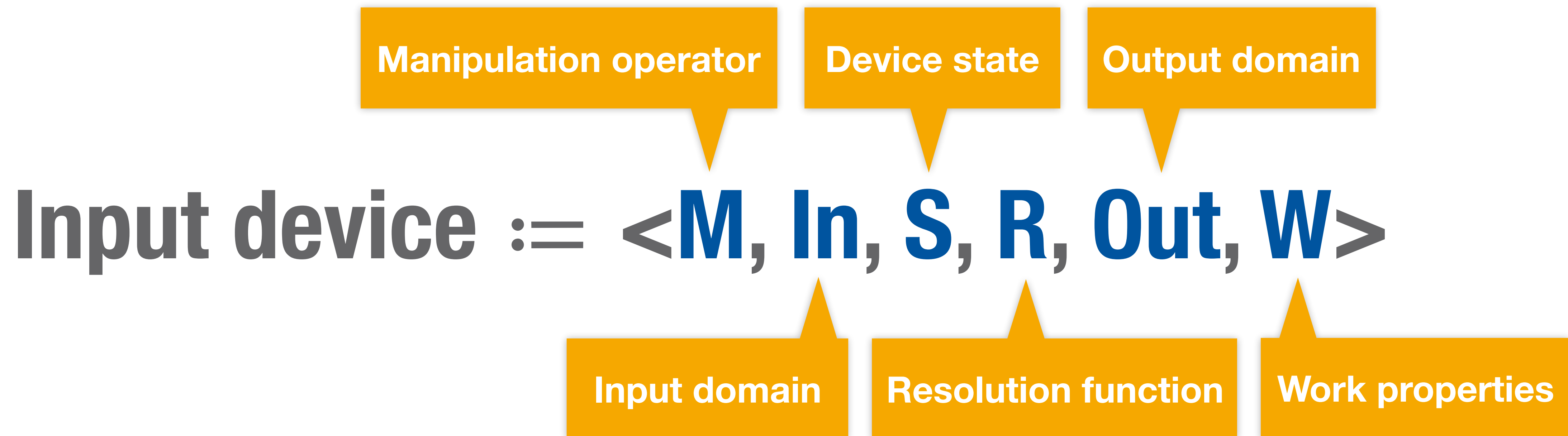
# Design Space of Input Devices



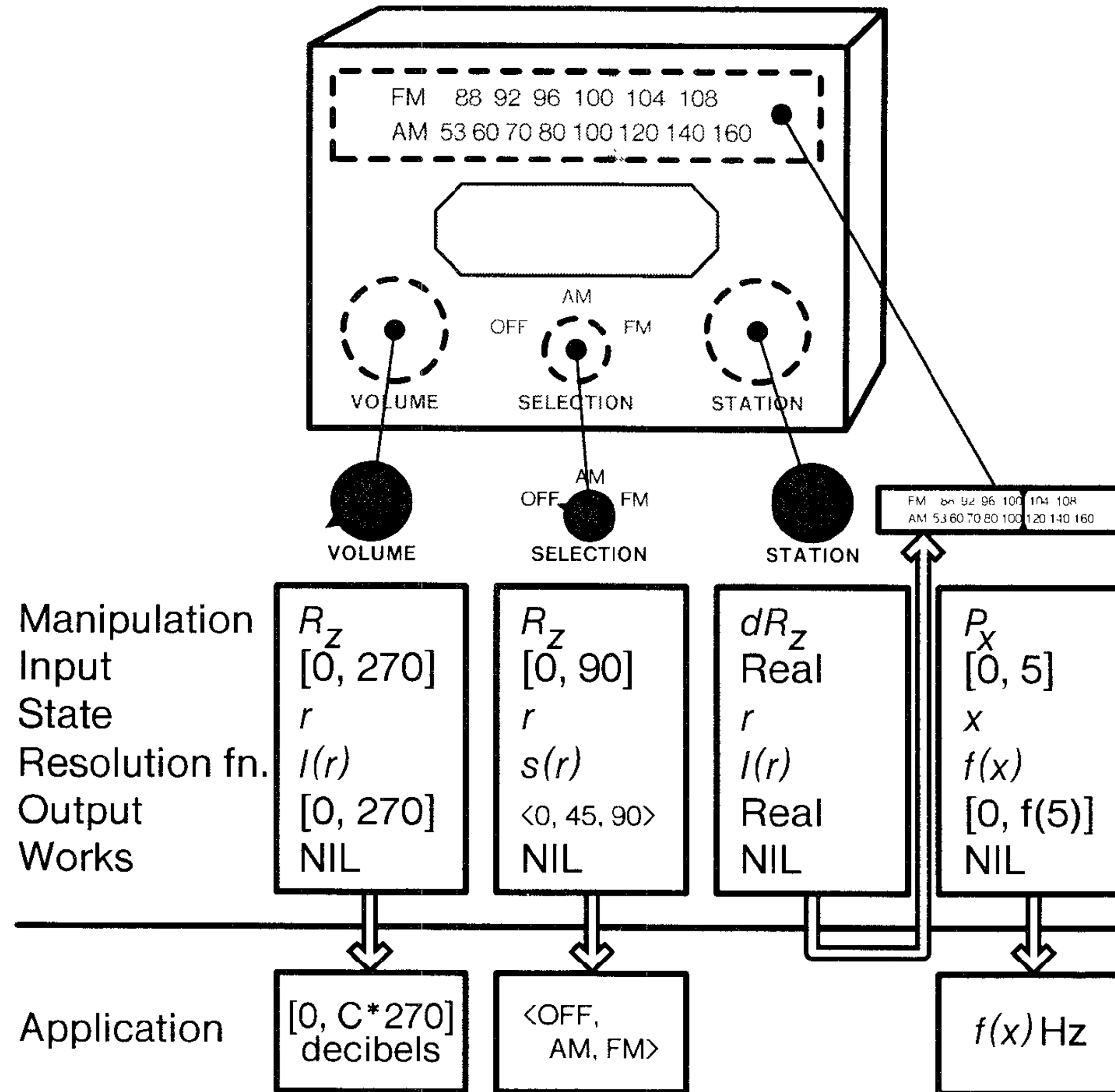
# Design Space of Input Devices

- Card, Mackinlay, Robertson 1991
- Categorization of input devices according to physical, mechanical and spatial properties
- **Why?**
  - Compare input devices
  - Identify new input modalities

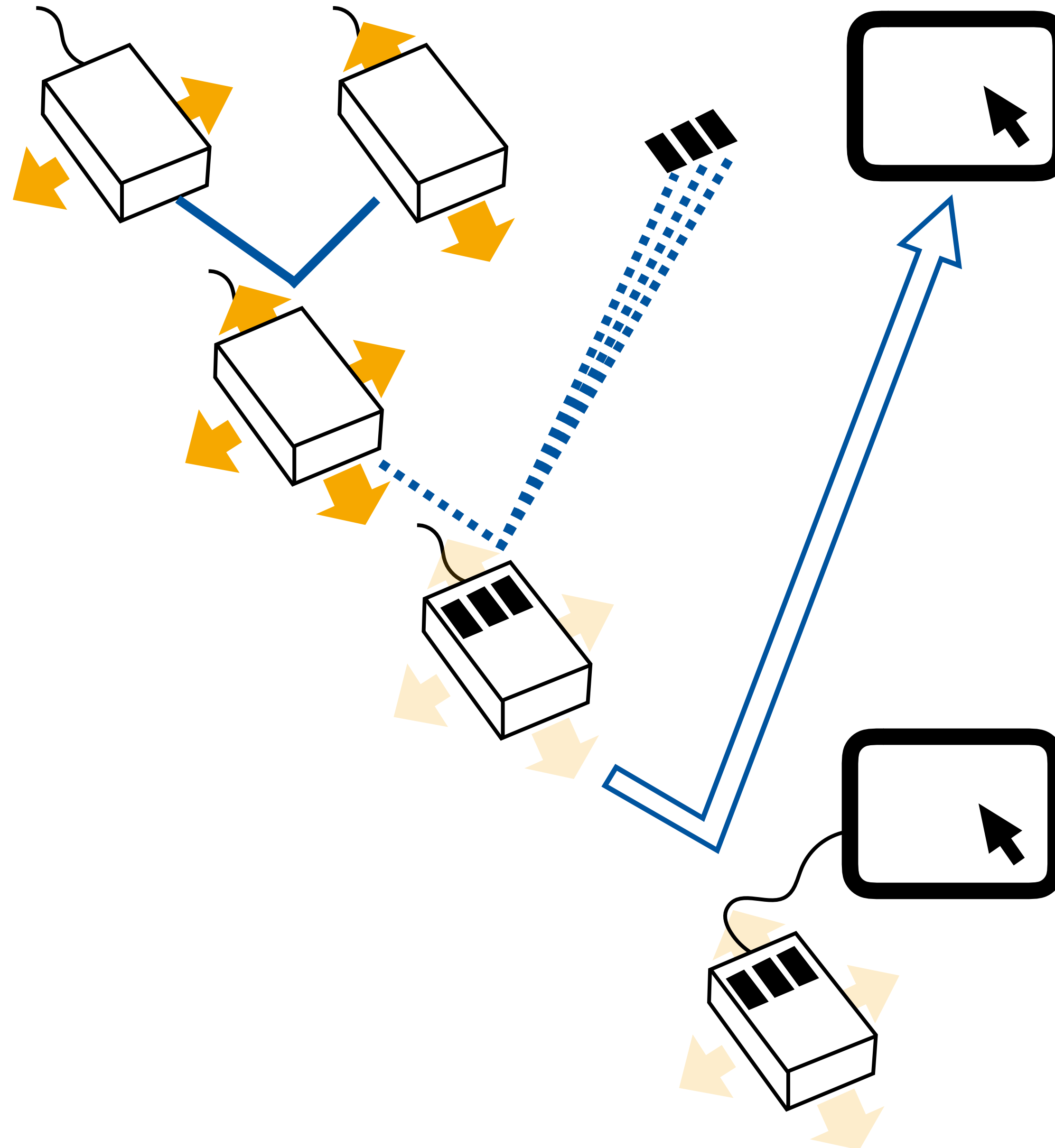
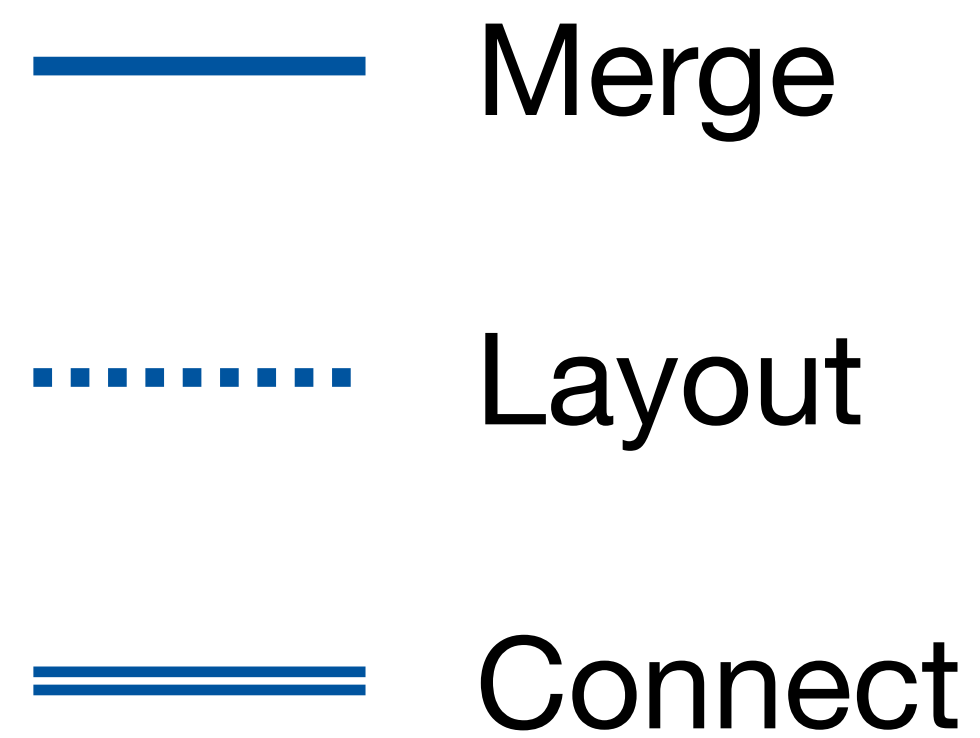
# Movement Primitives



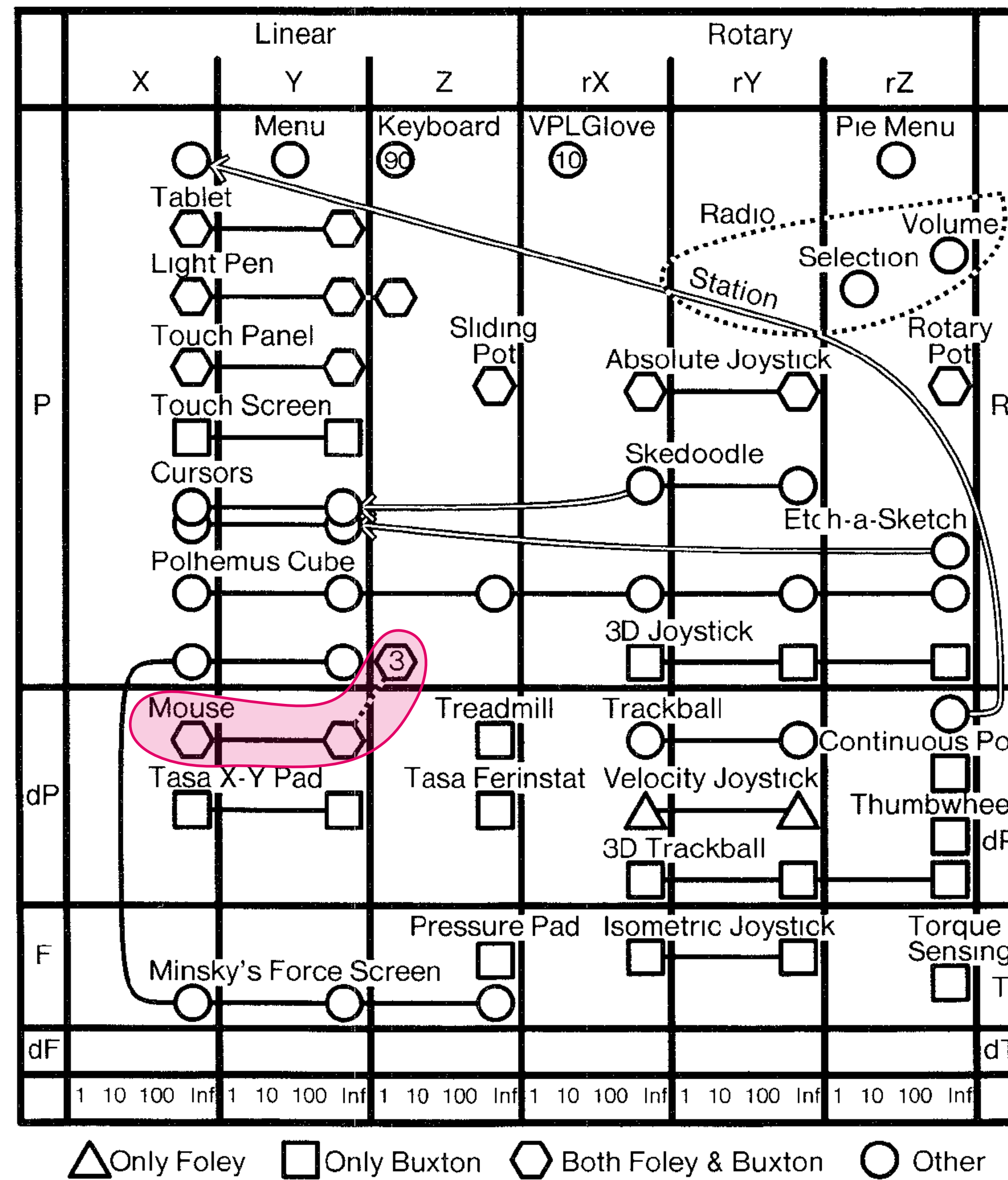
# Example



# Compositions







# In-Class Exercise

- Plot out the input capabilities of the Ferrari Racing Controller on the Card Design Space of Input Devices.
- The controller consists of a **steering wheel** with **8 buttons** and a **rotary switch** with 5 states, as well as **2 pedals**.
- Assume that the steering wheel can only have one full rotation.





# In-Class Exercise

		Linear				Rotar				
	X	Y	Z	rX	rY	rZ				
P			8							R
dP										dR
F										T
dF										dT
	1	Inf	1	Inf	1	Inf	1	Inf	1	Inf

1 switch with 5 states

1 steering wheel with an infinite number of states

8 buttons with 2 states each

2 pedals with an infinite number of states each



# Is This Space Complete?

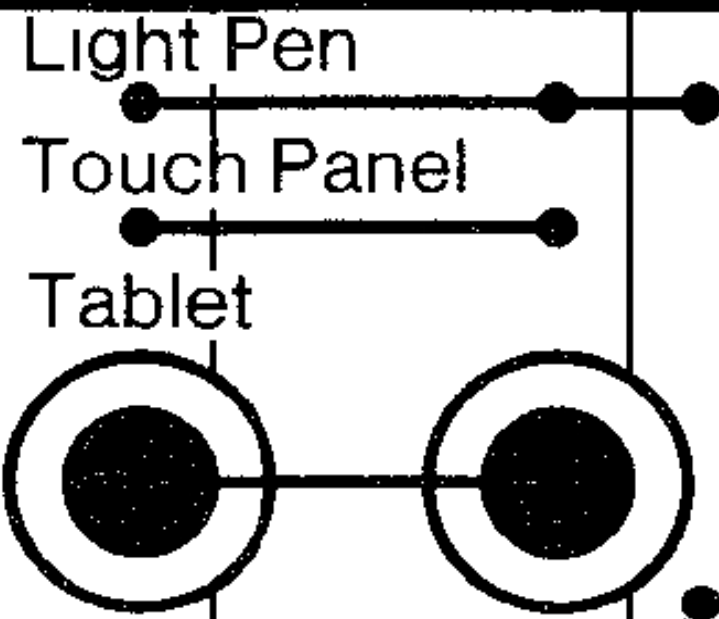


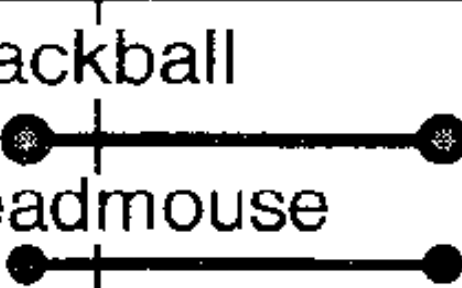


# Testing Points

- **Expressiveness** describes how precisely the meaning is conveyed
- For input devices, expressiveness suffers if  $|In| \neq |Out|$ 
  - $|In| < |Out|$ : Cannot specify all legal values
  - $|In| > |Out|$ : Can specify illegal values

# Testing Points

- **Effectiveness** describes how well the intention can be communicated

	Linear												Rotary												
	X				Y				Z				rX				rY				rZ				
P																									R
dP																									dR
F																									T
dF																									dT
	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	

# “Will I Get a Seat in This Class?”

## CHAPTER 3

# Window System Architecture

# Window Systems: Basic Tasks

- **Input handling**  
Pass user input to appropriate application
- **Output handling**  
Visualize application output in windows
- **Window management**  
Manage and provide user controls for windows



# Window Systems: Requirements

- **Independent** of hardware and operating system
- No noticeable **delays** (few ms) for basic operations, e.g. moving window, redrawing cursor
- **Customizable** look&feel for user preferences
- Input & Output in **parallel**
- **Multimedia** support: Graphics, audio, ...
- Support for various **input devices** and modalities

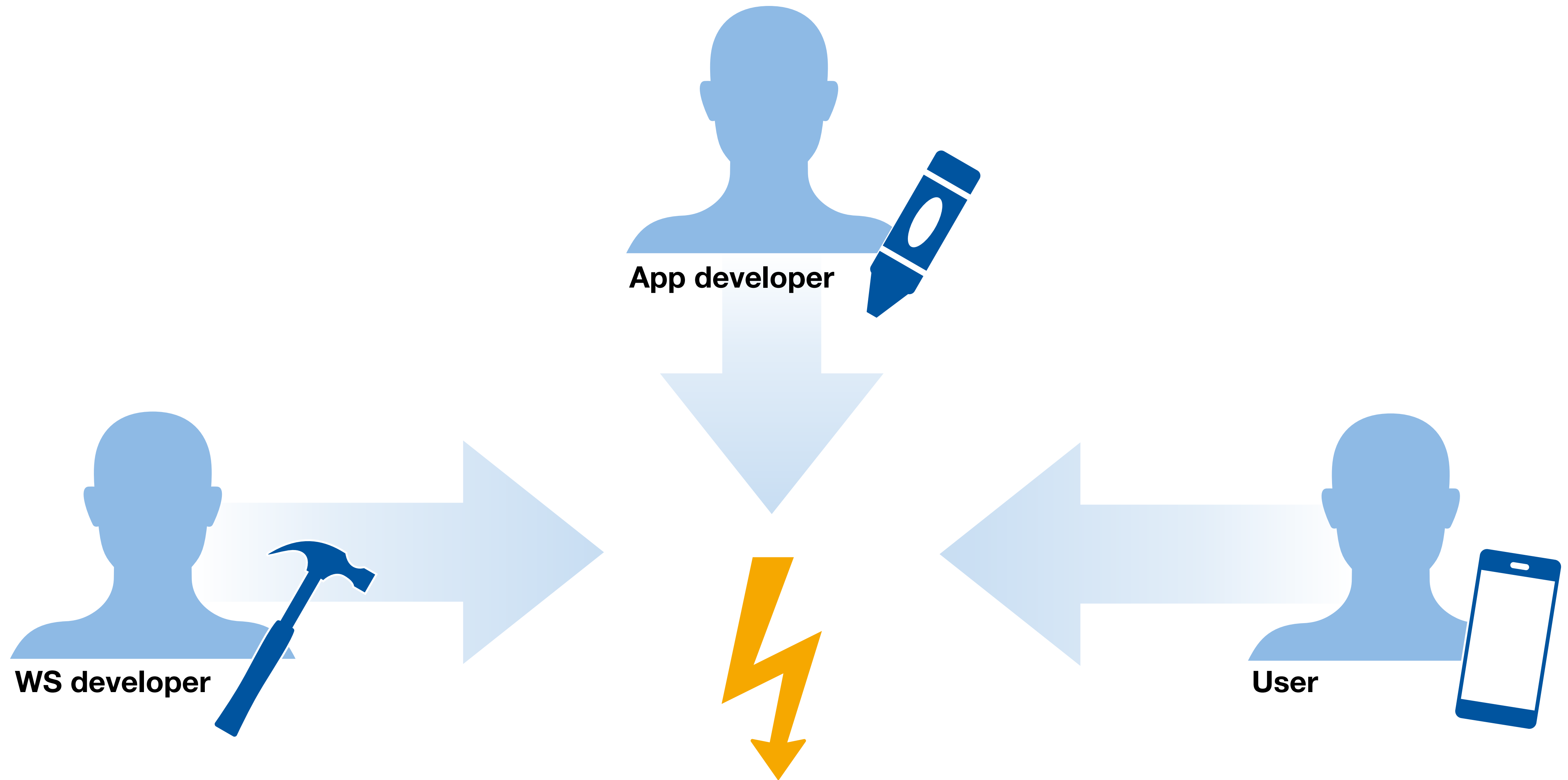
# Window Systems: Evaluation Criteria

- **Availability**  
Platforms supported
- **Productivity**  
For application development
- **Parallelism**  
External and internal
- **Performance**  
Usage of resources and latency
- **Graphics model**  
RasterOp vs. vector
- **Appearance**  
Look & Feel, exchangeable?

# Window Systems: Evaluation Criteria

- **Extensibility**  
In source code or at runtime
- **Adaptability**  
Localization and customization at runtime
- **Resource sharing**  
E.g., fonts
- **Distribution**  
Over network
- **API**  
Structure and comfort
- **Independence**  
Of application and interaction logic inside programs written for the WS
- **Inter-Application Communication**  
Copy & Paste, Drag & Drop

# Window Systems: Conflict





# Window System Architecture

