1. Is the situation shown at the start of the video the only time we need rate limiting?

Rate limiting is required for healthy functioning of a system. You might have rate limiting as a business use case (no more than 10 orders a minute) or to reduce lock contention (Imagine a ticket booking system).

Sometimes rate limiting is done to maintain one's SLAs (response times, failure rates, etc...). Rate limiting is also useful, as shown in the start of this video, to avoid a cascading failure.

2. Is a single rate limiting component used across the entire system? Isn't it a bottleneck and a single point of failure?

The rate limiter will be a distributed service which can be scaled horizontally. So all common use-cases can be handled here.

But a service might need more custom methods of rate limiting for itself. In that case, it can implement them itself. For example, a chat service might have complicated rate limiting logic and behaviour. This may not be compatible with the other services. In this case, the chat service implements an internal rate limiter.

3. How do you set the limit for each service?

Capacity estimation gives us a good number to start.

For example, if you have 3 machines with 2 GB RAM and you want them to be caches for 2KB profiles, you can have 3*2GB/2KB = 3 million profiles in the cache.

If your system has 10 million active users, and 90% can be served by cache, you have an average profile lookup time of 0.9 * memory_lookup_time + 0.1 * db_lookup_time.

That's about 0.9 * 0.01 ms + 0.1 * 1 ms = 0.009 + 0.1 ms = 0.11 ms per profile search.

So we should be able to serve 1/0.11 = 9 requests per ms. That's 9000 requests per second.

We then take a conservative estimate of 5000 requests per second for our load tests. After load testing, we can confidently say how many requests our service will be able to handle (with good response times).

4. Doesn't rate limiting add to the memory requirements and latency of each request?

Yes, rate limiting is an additional feature which requires design, maintenance and performance tuning. Since the advantages of rate limiting outweigh the drawbacks, it is worth it.

5. I have heard of terms like back pressure. What is it?

It's dynamically accepting requests based on how many you have left to process now.

Try this article and other links in the description for a detailed understanding: https://medium.com/@jayphelps/backpressure-explained-the-flow-of-data-through-software-2350b3e77ce7